DOCUMENT RESUME

ED 247 909 IR 011 259

AUTHOR	Aichele, Douglas B., Ed.; And Others
TITLE	Proceedings: Annual Microcomputer Conference (3rd, Stillwater, Oklahoma, November 18-19, 1983).
INSTITUTION	Oklahoma State Univ., Stillwater.
PUB DATE	Nov 83
NOTE	78p.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Guides - Non-Classroom Use (055) -- Reports - Descriptive (141)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; *Computer Oriented Programs; Computer Software; Elementary Secondary Education; Instructional Materials; *Microcomputers; Postsecondary Education; *Programing; Programing Languages; Teacher Education
IDENTIFIERS	Computer Uses In Education; *LOGO Programing Language

ABSTRACT

LOGO programming language and microcomputer applications for educators are the primary themes addressed in this conference collection of 2 keynote speeches and 32 presentation abstracts (with reference lists). The keynote papers are "LOGO and the Retraining of Teachers," by Kathleen A. Martin and "Should I Use LOGO?" by James E. Muller; the abstracts summarize presentations by 5 education administrators, 3 elementary teachers, 5 secondary teachers, 5 special educators, 6 postsecondary educators, and 8 teacher educators. Specific topics addressed include accountability, computing readability by microcomputer, using microcomputers for editing business letters and reports, PASCAL, building the high school computer curriculum, teaching programming, using the computer with gifted and talented students, learning disabled students and computer assisted instruction (CAI), evaluating software for CAI, preparing instructional still visuals using a microcomputer, CAI in three sensory modes for foreign language learning, computer science needs in Oklahoma, the role of microcomputers in a preservice elementary math methods course, computer literacy guidelines (K-12), and important considerations for teacher training in microcomputer technology. (Author/LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Douglas B. Aichele

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Table of Contents

Page

Preface..... i

Part I - Keynote Addresses

1.1 LOGO and the Retraining of Teachers
Kathleen A. Martin..... 1
1.2 Should I Use LOGO?
James H. Muller..... 9

Part II - Abstracts of Papers Presented

2.1 Education Administrators.....31

2.1.1. One School District's Response to Accountability for Student Progress in Reading and Mathematics - Martha Combs.....31
2.1.2. Computing Readability by Microcomputer - Karen Connel and Susan Mills.....32
2.1.3. Teachers and Computers Meeting the Challenge - Charles A. Hanna.....33
2.1.4. Microcomputer Use in Selected Small Oklahoma Schools - Richard Hunter.....34
2.1.5. Computerizing the Secondary School Curriculum: An Administrative Approach - Kara Gae Wilson.....35

2.2 Elementary Teachers.....37

2.2.1. Variables for Students Under Ten - James E. Evans....37
2.2.2. Implementing the "Impossible": One Computer and Thirty Kids - Jo Ann McEwen.....37
2.2.3. LOGO: Overview - Malcom Phelps.....38

2.3 Secondary Teachers.....40

2.3.1. Using Microcomputers for Editing Business Letters and Reports - Karen Forcht.....40
2.3.2. PASCAL or Is There Life After Basic? - Philip Knight.....41
2.3.3. Building the High School Computer Curriculum - Tony Lutman.....43
2.3.4. Teaching of Arrays with Applications - Paul Ohme and Lejeane Thomas.....44
2.3.5. Teaching Programming: An Initial Experience in Problem Solving - Thomas R. Rahlfs.....45

2.4 Special Educators.....47

2.4.1. Using the Computer with Gifted and Talented - Kay Sather Bull.....47
2.4.2. Computers in Classroom: Teachers of the Gifted - George Ann Guse.....48
2.4.3. Surviving Skills for the Teacher and Student in Mainstreaming - Sandra Haven.....49
2.4.4. Learning Disabled Elementary Students and CAI - KimberLee Powell-Rahlfs.....50
2.4.5. Evaluating Software for Special Education - Gay C. Neuberger.....51

2.5 Postsecondary Educators.....53

2.5.1. Mega Enthusiasm with Micros - Brenda Cantwell.....53
2.5.2. Microcomputers for Teachers: A Junior College Course - Joyce S. Friske.....54
2.5.3. Preparing Instructional Still Visuals Using a Microcomputer - Elton Li and Daryll Ray.....55
2.5.4. Bringing Microcomputers to the Faculty, Staff, and Students of a Small University - Bill Oldham.....56
2.5.5. Using a Time-Sharing Computer System in the Classroom: Large Screen Projection Television as an Interactive Teaching Aid - Charles D. Rorie....57
2.5.6. Computer-Assisted Instruction in Three Sensory Modes for Foreign Language Learning - Harry Wohlert..57

2.6 Teacher Educators.....59

2.6.1. Computer Science Needs in Oklahoma: Demands Exceed Supply - Richard Alexander, Hal Jenkins, Charles White.....59
2.6.2. Introduction to Computer Literacy on a Networking System - James Bowen.....60
2.6.3. The Role of Microcomputers in a Preservice Elementary Mathematics Methods Course: Implications for Teacher Education - Raymond Brie.....60
2.6.4. A Case for Teachers to Acquire Computer Literacy - Trenton Cooper.....62
2.6.5. Jousting with Jargon: Suggestions to Enhance Communication Between Computerists and Traditional Classroom Teachers - Jean Fincher.....64
2.6.6. Computer Literacy Curriculum Guidelines, K-12 - Martha Kay Hedley.....65
2.6.7. Setting the TEMPO for Instructional Computing Courses: An Organized Set of Teaching Methods for Computer-Related Skills - LaJeane Thomas.....66
2.6.8. Who, What, When, Where and How: Important Considerations for Teacher Training in Microcomputer Technology - Mary M. Worth.....68

Preface

The Department of Curriculum and Instruction at Oklahoma State University sponsored the Third Annual Conference on Microcomputers in Education on November 18-19, 1983. As with the previous microcomputer conferences, sessions were provided specifically for educational administrators, elementary teachers, secondary teachers, special educators, postsecondary educators, and teacher educators. The purpose of these Proceedings is to provide abstracts of the papers presented and to encourage sharing ideas in the future.

The main themes for the Third Annual Conference on Microcomputers were LOGO and microcomputer applications for educators. The keynote speakers were Dr. Kathleen A. Martin, University of Dallas, and Mr. James H. Muller, Young People's LOGO Association; their entire speeches are included here. We extend our sincere appreciation to Kathleen and Jim for their contributions to making this conference a real success.

The response to the call for papers to be presented at the Conference was tremendous. Consequently, we were able to select some excellent papers for each of the program strands. After careful review, each session paper was abstracted in an attempt to highlight that session. While it was not possible to reprint the session papers, the references cited by the presentors have been included. We extend special appreciation to the presentors who willingly shared their ideas and submitted papers.

No conference like this could have been possible without the untiring efforts of faculty and staff. The Conference Coordinator, Helen N. Cheek

(Curriculum and Instruction), deserves very special thanks. She worked diligently to build upon our previous conferences and incorporate new ideas into this one. Appreciation is also extended to the members of the Planning Committee: Jim Bowen (Curriculum and Instruction), N. Jo Campbell (Applied Behavioral Studies in Education), Joyce Friske (Curriculum and Instruction), John Gelder (Chemistry), Michael Kerr (Applied Behavioral Studies in Education), Constance Lawry (Education Extension), Brenda Lomenick (Curriculum and Instruction), Malcom Phelps (Curriculum and Instruction), and Susan Roth (Curriculum and Instruction).

Finally, I wish to express appreciation to Peggy Disel, Department of Curriculum and Instruction, for her assistance in preparing these Proceedings.

Douglas B. Aichele, Head
Department of Curriculum & Instruction
Oklahoma State University

PART I

KEYNOTE ADDRESSES

1.1 LOGO and the Retraining of Teachers

Kathleen A. Martin, University of Dallas

Many of the questions about microcomputers which currently absorb school people are not the significant questions. Unfortunately the American penchant for quantity and speed encourages technological concerns rather than the pedagogical concerns more appropriate to education. All too often questions of hardware preempt questions of software which preempt questions of teaching and learning. If the microcomputer is to make any real difference in the effectiveness of educational efforts, then more educators must return to the significant questions. Foremost among these questions are (1) What is it that we want our children to learn through the use of microcomputers? and (2) How can we as teachers facilitate that learning?

There are, of course, a number of answers to both questions. Personally, I would encourage that microcomputers be used to help children encounter important ideas and to think more clearly about those ideas. I would suggest that teachers can facilitate this encounter by asking appropriate questions of students at the appropriate times. While these answers may seem simple and obvious, they apparently are not. Otherwise there would not be so many people at conferences like this who are asking what to do with microcomputers and how to do it.

Part of the problem is the scarcity of software through which important ideas can be encountered. For years now we have been promised that the good software is coming. I, for one, am not holding my breath. My prediction is that good software will become readily available only when a lot of good teachers become good programmers. Offhand, I don't know many good teachers who have the time to become good programmers.

Another part of the problem lies in the choice of computer languages. The vocabulary and syntax of most computer languages are formidable, thereby making them inaccessible to the majority of elementary school age children. There is, however, at least one language whose design was intended to make it easy to communicate with the computer. The language is called LOGO and was authored by Seymour Papert of MIT. The speed with which LOGO is gaining popularity across the country (and internationally) suggests that it is helping address the significant educational questions.

In describing LOGO, Papert says that the language offers "objects-to-think-with." I would further suggest that LOGO offers objects-with-which-to-ask-questions." What I would like to do now is (1) to demonstrate some of the important ideas that emerge in working with LOGO, and (2) to describe how appropriate questions can encourage and cultivate those ideas. Because of the limitations of time, I am going to work with only the Turtle Graphics dimension of LOGO. On one level our content will be geometry (what we are thinking about), while on another level our content will be processes of conceptualization (how we are thinking about what we are thinking about).

I will assume that you have little or no experience with LOGO. Consequently, we will begin with a simple project; we will make a square. The Turtle (or the object with which we will be thinking) is a benign computer-controlled cybernetic creature that initially resides in the center of the screen (Figure 1).

Figure 1

Given the Command FD 30, the Turtle will advance 30 turtlesteps in the direction it is pointing (Figure 2).

Figure 2

A series of seven more commands will complete the square: RT 90, FD 30, RT 90, FD 30, RT 90, FD 30, RT 90 (Figure 3).

Figure 3

Here the question to ask a child is "Do you see a pattern?" If the child is able to identify the pattern, then he is taught a shortcut. That's what mathematical formulas are all about. In the case of the square, the shortcut is REPEAT 4[FD 30 RT 90].

The next question might be "What can you build with a square?" There are any number of possibilities. A window (Figure 4) could be built from a square: REPEAT 4[SQUARE RT 90]. Rotating a square could produce a fan (Figure 5): REPEAT 12[SQUARE RT 30].

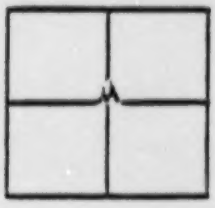


Figure 4

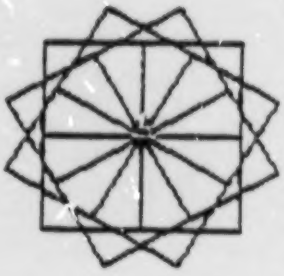


Figure 5

Notice how the window and the fan are more complex concepts built upon an understanding of the simpler concept of a square.

Another geometric form worthy of investigation is the circle. If you wanted to teach children how to write a program to generate a circle, you might first ask them to describe how to make a circle by walking. Eventually they would tell you that walking in a circle involves taking a small step forward and then turning a little bit and that the process should continue until you return to the place where you started. In LOGO the program would read as follows: REPEAT 360 [FD 1 RT 1]. The study of circles offers a wonderful opportunity to ask good questions: How do you make the circle smaller? What affect does changing the REPEAT number have upon the RT angle? What affect does it have upon the FD distance?

This latter problem of circle-making points to the importance of having children work off the computer as well as on it. In the case of circles it is helpful to use the child's intuitive sense of his own body geometry. Some other off-computer activities include work with mirrors and tessellations, paper-folding, and string designs (see references). The insights that a child gathers through the off-computer activities are then translated to the computer screen. For example, paper-folding activities help a child to realize that the central angle of any regular polygon is the same measure as the exterior angle of the polygon. Since the sum of the central angles of any regular polygon is always 360 degees and since the central angles are equal in size, the measure of the central angle is 360 degrees divided by the number of sides of the polygon. Thus the central angle of a pentagon equals 360 degrees divided by 5 or 72 degrees. The exterior angle, or the turning angle of the Turtle, is the same. The child quickly learns to construct a LOGO program that enables him to construct any regular polygon.

```
TO POLYGON :NUMBER
REPEAT :NUMBER [FD 30 RT 360/ :NUMBER]
END
```

Using numbers from 3 to 12, the child can generate the designs in Figure 6. What is a significant question that can be asked here?

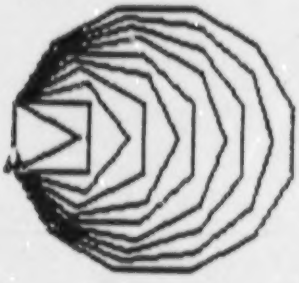


Figure 6

Once polygons are well in hand, the child can proceed to spirals. Here again body geometry is helpful. A square spiral is easy to walk and easy to describe.

```
TO SQUARE-SPIRAL :SIZE :LIMIT :ANGLE
IF :SIZE > :LIMIT [STOP]
FD :SIZE
RT :ANGLE
SQUARE-SPIRAL :SIZE + 4 :LIMIT :ANGLE
END
```

If a size of 1, a limit of 60, and an angle of 90 are used, Figure 7 results. Changing the angle to 120 degrees generates a triangular spiral (Figure 8). While an angle of 72 produces a pentagonal spiral (Figure 9).

Figure 7

Figure 8

Figure 9

When spirals are connected to spirals, some really interesting questions can be raised.

```
TO CONNECT-SPIRALS :NUMBER :SIZE :LIMIT :ANGLE
REPEAT :NUMBER [SQUARE-SPIRAL :SIZE :LIMIT :ANGLE RT :ANGLE FD 40]
END
```

If a number of 4 and an angle of 90 are used, square spirals will be connected. A limit of 60 keeps the spirals from getting too big and going off the screen. The only variable that changes is the size. Figures 10, 11, 12, and 13 show what happens with sizes of 1, 5, 9, and 13.

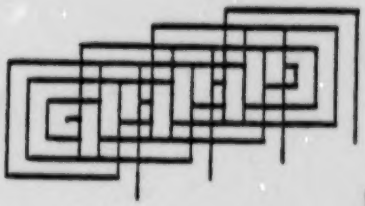


Figure 10

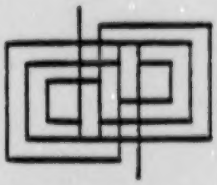


Figure 12

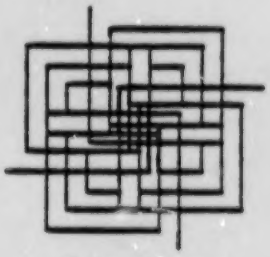


Figure 11

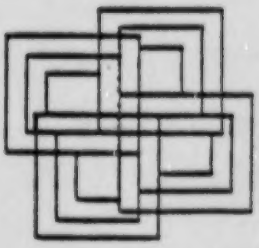


Figure 13

What would happen if a size of 17 was used? What would happen with the in-between sizes? How many different patterns are there? Why these particular patterns? As the geometry becomes increasingly complex, so do the questions. What is so exciting about these more complex questions is that they seem to open even more questions.

Let's look at another kind of spiral called an inspiral. The figures below are the consequence of the following LOGO program.

```
TO INSPI :ANGLE
FD 10 RT :ANGLE
INSPI :ANGLE + 10
END
```

Figure 14

Figure 15

Figure 16

Figure 17

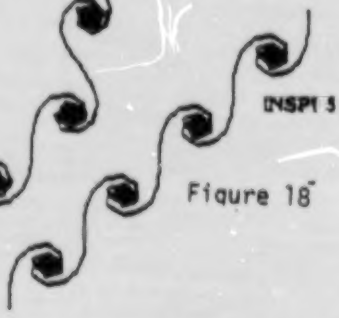


Figure 18

Figure 19

Figure 20

Figure 21

Figure 22

What do you notice about the figures? Why doesn't Figure 18 have a partner? You probably observed that the program would run "forever" unless broken. What would you have to know in order to make the Turtle stop at the place of its beginning? Do you see any resemblance between the inspirals and the spirals that we did earlier?

I would hope that my point is beginning to be made through these example programs. Computers can certainly be used to help children think about important mathematical ideas and to better understand their own processes of conceptualization. They can also be used in some very ineffective and mundane ways that fail to argue their worth. The difference is determined by our choices of software, including languages like LOGO, and by the quality of the questions which that software encourages us to ask. Doubtless we will all need to undergo some retraining and some mental retooling to meet the challenges of computer technology. However, not all that we need to learn is new. Good teachers have always asked good questions. Computers simply demonstrate that attention to significant questions is more important than ever.

1.2 Should I Use LOGO?

James H. Muller, Richardson, Texas

"Personal computers should be made as readily available to a child as pencils," observed Dr. Seymour Papert before a group of educators at the University of Dallas, Texas, in November of 1981. "The pencil is to scribble, to doodle, to draw, to write, to work sums, and to chew on. It is used for illicit notes as well as for official assignments. I see the computer in the life of the child as equally ubiquitous and equally versatile. I also see it as equally personal. Children own pencils; they are not intimidated by them. This should be equally true of the child's personal computer."

Dr. Papert is one of the world's leading authorities on how children absorb knowledge and how best to develop their learning skills. He is also one of a large number of experts who recognize the enormous impact computer science is having on the educational process, both in the classroom and in the home. With his associates at the Artificial Intelligence Laboratory at MIT, he spent twelve years developing a unique philosophy of education which has been implemented through the computer language, "Logo."

Logo has been characterized in magazines, books, educational conferences, and various other outlets as everything from the saviour of the education process to just another fad. Obviously it is neither. Like the computer, it is nothing more or less than a learning tool. It's value depends on its effective use. But before this value can be truly appreciated, look at the environment into which it is being introduced.

Today, even with computers already a pervasive part of our everyday world, electronic technology is far less forbidding to the average child than to his parents. Adults are often totally bewildered by the on-rushing pace of technology. To the young person, the computer is nothing more or less than a fascinating tool. But parents tend to blame an endless variety of problems on "the computer."

Think about it for a moment! Is it correct to blame the pencil for a writing error? Is the fork at fault for putting some bad-tasting food in the mouth? Is the TV to blame for poor programs? Is, indeed, the computer to blame for an incorrect bill. But isn't it a lot easier to blame these inanimate objects or the technology involved?

Is the technology truly at fault?

People have been abusing technology ever since a caveman found that his favorite rock would split his enemies' heads with the same ease with which he split a coconut. The computer is no exception, offering a wide variety of opportunities for use and abuse.

Many of these opportunities for abuse stem from lack of understanding. And that in itself, is quite understandable. Most technologies which we readily accept as part of our everyday lives, have risen slowly. The airplane, the radio, the automobile, and the telephone evolved over many years. However, the computer has been sprung on us over very few years. No one has had time "to grow with it."

The single-chip microcomputer that powers a child's electronic toy, the same one which can sit on a baby's fingernail, would have filled a small room if built twenty years ago. And, it would have cost well over $10,000. If one of today's more powerful personal computers could have been built back then, it would have taken the Empire State Building to house it and the Hudson River to cool it.

Parents and teachers of today have grown up with this exploding, forboding, and very confusing technology. The common perception has justifiably been that computers require trained technicians and expert programmers to operate them. Computers have been strange machines hidden in secret air-conditioned rooms, accessed only by the elite few. How quickly this has all changed.

Personal computers are now sold by the millions. Computerized toys and appliances have been sold by the tens of millions. Automobiles, airplanes, burglar alarms, smoke detectors, and numerous other household devices are now computerized. And all of this has come within less than ten years. Is there any wonder why some people feel confused and intimidated?

When today's young people reach adulthood, their environment will have been computerized to the degree that it will be totally accepted as the way the world IS! For this reason, it is becoming increasingly important for children's education to incorporate familiarity with, and an understanding of, a technology which will be as much a part of their surroundings tomorrow as pencils and paper are today. Thus, it behooves today's parents and teachers to look beyond their conditioning, their fear of microcomputer technology, to examine how this most exciting learning tool can be used effectively.

The typical use of the computer has been with talented and gifted students who use preprogrammed software as drill and practice for math and other subjects. They use the computer for everything from reliving the Civil War to simulating the liftoff of the Space Shuttle or calculating the gravitational pull of Jupiter's moons.

But this type of application is not without its problems. Rather than learn to think for themselves, students must rely on the creativity and teaching skills of some distant and unknown programmer who may or may not have any classroom experience. Indeed some obviously have little concept of

education. While Computer-Assisted Instruction has some value in proper educational context, such uses tend to limit the true potential of the computer. Many CAI programs relegate the computer to the role of simple audio visual aid.

An alternative to CAI is to teach young people to use their own creativity and imagination to develop their own programs. But until recently, teaching young people to program has been as cumbersome as some of the educational software readily available. Words such as INIT, CHR$, MID$, ABS, GOSUB, and SYNTAX ERROR are not part of a child's normal vocabulary. Coupled with the complexities of using such high level languages, young people tend to find it extremely difficult to fully explore the potential of the computer. They tend to merely memorize the commands and their use. But they have little concept of how to effectively use the language or the computer. They can duplicate existing materials but often lack the creativity to develop original programs.

This is reminescent of those students who see mathematics as nothing more than a series of tables and formulas to be memorized. They fail to see math as a universal language for problem-solving. They gain knowledge, but have no concept of how to use it.

Some of the more innovative educators have used the computer and other electronic learning aids to help the learning disabled and the so-called "slow learners." Programs have been individualized for the specific needs of each child so that each is continually challenged and continues to move at their own pace. The child, in turn, develops a personal one-on-one relationship with the computer, receiving immediate feedback and reinforcement on their level.

More importantly, the child is not embarrassed in front of peers or the teacher if a wrong answer is entered. The child merely tries again. As children experience this form of immediate success with a correct answer, they

are encouraged to continue to learn, even to excel. Slow-learners gain valuable self-confidence in their approach to problems and decision-making. They will often become more creative because they use the computer to deal with more interesting, real-life situations, free from the cumbersome nature of calculation and administrative tasks which otherwise can be quite overwhelming.

With the introduction of Logo, this process is even further enhanced. The student is able to become immediately involved in a unique, computer-based learning environment where mathematics, language arts, logic, and other areas of traditional learning can be absorbed in an instinctive, natural fashion, much as a child learns to speak or experience the wonders of a butterfly--or a turtle.

The basic philosophy of Logo is one of self-directed explorations through which children learn by teaching. Rather than have the computer "program" the learning process, children create their own learning environment wherein they learn by teaching (or programming) the computer.

But Logo is far more than just a learning language. It can serve the needs of all ages and experience. It is much like the game of chess. A six-year old can learn the basic moves of chess, but elderly masters still seek to learn more about the strategic manipulation of opening, middle, and end-game procedures. Like Logo, it has no threshold and no ceiling. The flexibility and expandability of the game is limited only by the imagination.

Young people are traditionally introduced to Logo through its most popular feature, Turtle Graphics. When the language was first implemented on a large computer, an electromechanical device was developed which would move around the floor on command. Pens could be raised and lowered to allow children to draw pictures using a separate simplified keyboard. The dome-shaped device resembled a turtle, and thus the term, "turtle graphics" evolved.

With the growing use of personal computers in the classroom, the Logo language has been compacted to fit within the limits of microcomputer memory. The Turtle no longer roams the floor drawing on paper. Now it is a cybernetic pet residing on the computer screen. Most commonly, the Logo Turtle appears as a small triangle with the smaller angle of the triangle indicating the direction the Turtle is facing. More recent implementations, such as Atari Logo (TM), use a small graphic Turtle with which young children, expecially, can more readily identify.

Logo is a friendly, conversational language which uses plain English statements as opposed to the symbolic code of BASIC and some other common high level languages. Using very simple commands, even preschoolers can learn to control the Turtle, moving it FORWARD, BACK, LEFT, and RIGHT. They also find they can have PENUP, PENDOWN, PENERASE, AND PENREVERSE. Children discover the actions of the Turtle through their own controlled explorations and discoveries. They quickly develop a sense of number relationships by telling the Turtle to move FORWARD different values. They also begin to develop a sense of turns and angles as they teach the Turtle to turn.

From their preliminary explorations, children learn the logical processes required to put a common geometrical shape on the screen. For example, they quickly grasp the sequence of commands required to draw a square.

```
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
FORWARD 50
RIGHT 90
```

From this basic shape, children will "see" all sorts of opportunities. For example, a traingle on top of the square created a roof for a house. Thus, they add:

```
FORWARD 50
RIGHT 120
FORWARD 50
RIGHT 120
FORWARD 50
RIGHT 120
```

The only problem with this procedure is that the roof appears inside the house. But this is not necessarily wrong--it is merely a bug that has to be corrected. Indeed, it is the "mistakes" in Logo programming that provide the greatest learning opportunities.

To complete the house, the Turtle must be told to move to the top of the square and then turn 30 so it will sit properly as a roof.

```
REPEAT 4 [FORWARD 50 RIGHT 90]
FORWARD 50
RIGHT 30
REPEAT 3 [FORWARD 50 RIGHT 120]
```

Now the triangle is properly positioned on top of the square to present a rudimentary picture of a house.

Having drawn the house, the computer can be "taught" to draw any number of houses by defining a new Logo command, "House." This time,

```
TO HOUSE
REPEAT 4 [FORWARD 50 RIGHT 90]
FORWARD 50 RIGHT 30
REPEAT 3 [FORWARD 50 RIGHT 120
END
```

By entering the new command "House," the triangle is drawn on top of the square.

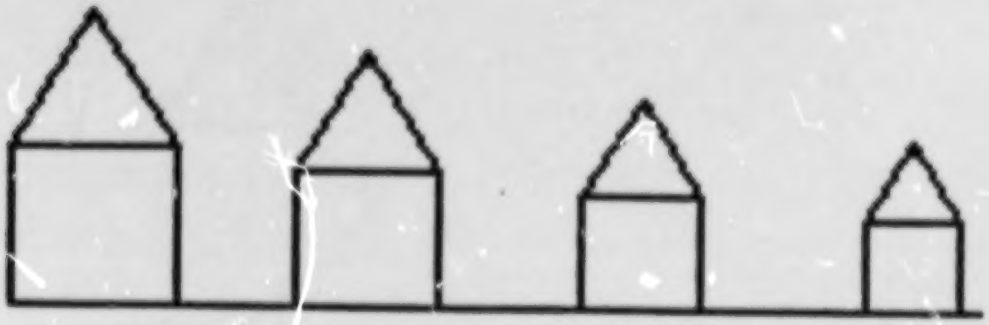
However this is only the beginning of the power of Logo. Now the command can be used to draw a suburban community.

```
TO SUBURB
SETX -100
HOUSE
SETX -50
HOUSE
SETX 0
HOUSE
END
```

Now, when the new command, SUBURB, is entered, three houses are drawn. But even this is a bit cumbersome. To draw a SUBURB with different distances and dimensions requires the procedure

```
TO HOUSE :DISTANCE
REPEAT 4[FORWARD :DISTANCE RIGHT 90]
FORWARD :DISTANCE RIGHT 30
REPEAT 3[FORWARD :DISTANCE RIGHT 120]
SETY 0 SETHEADING 0
END

TO SUBURB :DISTANCE :SPACE
SETX :SPACE
HOUSE :DISTANCE
MAKE "SPACE :SPACE + 60
MAKE "DISTANCE :DISTANCE - 5
WINDOW
SUBURB :DISTANCE :SPACE
END
```



Both House and Suburb have now been redefined to include the variables :DISTANCE and :SPACE. DISTANCE represents a side of the house, while SPACE represents the space between houses. When SUBURB is entered now, along with numerical values for DISTANCE and SPACE, series of houses will be drawn, each smaller than the preceding house.

There are, indeed, some very powerful mathematical and logical concepts involved in the development of these procedures. But, of far greater importance, consider what has occurred within the children's minds as they discovered these and other procedures.

From the simplest "known"--drawing a square--they moved through progressive stages to develop a suburban community. They took a problem, broke it into "mind-sized bytes" and reassembled these into a structured, procedural solution to the problem. In the process, they have explored the concepts of Cartesian Coordinates, geometry, and algebra. The primary grade youngsters who go through similar exercises obviously don't know these concepts by those names. But they quickly develop an instinct for these and other mathematical and logical complexities.

The "instinct" stems from the natural curiosity of children as they explore different options. They develop a far more rational and realistic sense of self-reliance by working through the structure of a problem. They quickly find that there are no mistakes in Logo, merely opportunities.

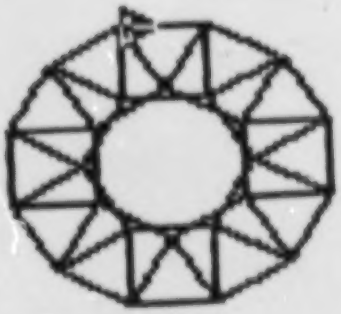
Not long ago, a young lady was developing her own Suburb. She had completed the House with the Turtle at the upper lefthand corner of the Square.

"What would happen if I entered House now?" she asked.

"Try it and find out!"

The result was another House growing out of the roof of the first. But this fascinated her. And so she entered House again, and a third House appeared

"Let's see? If I repeat House often enough, that should create an interesting pattern."



"That looks like a sun."

"No, it's more like an Indian design."

The young programmer came up with the definitive answer, however. She ended the discussion with the comment, "No, it looks like a wheel."

From the first "mistake," a new procedure evolved which was appropriately named Sportscar.

Logo is, of course, far more than just Turtle Graphics. It can be used in virtually any learning situation: geography, language arts, social studies, science.

Here is a simple procedure (Tables I and II) which allows children to construct sentences using various parts of speech. The sentences are totally random and can end up to be quite humorous. But within this humor comes structured and procedural thinking--first to properly define the parts of speech, and then to correctly structure the sentence in which they can be used. An interesting classroom exercise would be to define other parts of speech and then construct new types of sentences, changing the sequence in the "Grammar" procedure.

It is the fun of these types of experiences that help even very young children develop a unique sense of confidence at the computer. Not only do they control their own learning environment, they develop keen perceptual skills. Unfettered by the conditioning adults have lived through, children are able to make the most of their imaginations and their creativity.

In the summer of 1981, some young children ranging in age from six to ten matched their new-found conceptual skills with those of experienced engineers who spent their days at the drawing board. Each group was asked to draw odd-sized angles freehand on blank sheets of paper. One accommodation was made for the children, however. Rather than being asked to draw a 147-degree angle, they were asked to draw a turtle's trail showing 147 Turtle Turns. Measured with protactors, the youngsters won hands-down. Through their experience with Logo, they had developed an intuitive sense, an acute ability to visualize spacial relationships.

The learning experiences of these primary grade youngsters go far beyond mere programming, far beyond some of the simplistic definitions of "computer literacy." These children learned to think. There are those young students who have used Logo for several years only to become brilliant typists. They know the commands and can enter these to create brilliant graphic displays. But behind the flash, there is not one creative procedure, not one original thought.

Herein lies the value of Logo. Used effectively, it can help children develop analytical and creative skills. To accomplish this, children need to go beyond the abstraction of the computer screen. They need the kinetic experience of feeling the concepts with which they are working. They learn best through the multi-sensory experiences of touching, hearing, and seeing.

In addition, most children also need the motivation to fully explore their learning environment. Some will naturally gravitate to programming and "keep at it" until the plug is pulled. Some others need very well-defined directions and progressive steps. Most young people, however, fit in between these two extremes. Left on their own to explore the computer, they tend to drift. They tend to rationalize the unknown away and confine themselves to the security of the known.

This is the real challenge to parents and teachers. Should children be allowed to drift? Should they be allowed to explore the computer on their own, at their own pace, without some objective? Is there a better way?

Rather than concentrate so much on "computer literacy," why not give children the opportunity to explore their true creative potential? Why not give them the opportunity to better know themselves, to become "self-literate?"!

Today's adults may not understand computers or even feel comfortable with them. They need merely apply the ageless instinct of parents so aptly described by D. H. Lawrence in "Fantasia of the Unconscious."

> "And this is the way to educate children: the instinctive way of mothers. There should be no effort to teach children to think, to have ideas; only to lift them and urge them into dynamic activity. The voice of dynamic sound, not the words of understanding. Damn understanding. Gestures and touch, and expression of face, not theory. Never have ideas about children--and never have ideas FOR them."

There is distinct value in giving children the opportunity to define their own learning environment at the computer. But they should have the opportunity to excel, to make the most of their creative energies. These energies need to be channeled through unobtrusive guidance.

Children need the opportunity to design and assemble their own "cart and horse." Adults can then become the "dynamic carrot" to guide that cart, to channel those bundles of creative energy so they have the opportunity to learn.

The story goes that you can lead a horse to water but not make him drink. However, if you create the environment in which the horse becomes very thirsty, he will, indeed, drink. You can teach a child, but not "learn" him. However, as you can create an environment in which the horse becomes very thirsty for water, an environment can be created in which the child becomes thirsty to learn. This is the environment of the "dynamic carrot."

Of more practical interest is the far more typical problem of one teacher, one computer, 30 students, and a 45-minute class period. How can these elements be effectively combined?

A most productive answer is through the development of learning centers, small groups of children, each exploring on and off computer activities all designed to exemplify the same concept of Turtle Graphics, geometry, logic, or another subject. By exploring a problem on graph paper, through body geometry and other kinetic exercises, and then on the computer, children gain a more intuitive sense of the complexities with which they are working. They learn

how to break many types of problems into "mind-sized bytes" and to structure their solutions in logical procedures. They begin to sense the utility of Logo and the computer as effective "tools of the mind."

It has been said that Logo is a children's language. Maybe that's because it brings out the child in its users. It is a unique language for text manipulation and list processing. It is also a flexible and friendly general purpose programming language. It offers experienced physicists the opportunity to graphically explore the theory of relativity and three-year-old children equal opportunity to explore the cybernetic world of the Turtle.

For a look at the applications of Logo, consider what Logo might do for the physically handicapped. Think of a utility that might offer a single keystroke language to communicate through a printer or other device.

```
TO A
.PRINTER 1
PRINT [JAMES H. MULLER, PRESIDENT]
PRINT [YOUNG PEOPLES' LOGO ASSOCIATON]
PRINT [1208 HILLSDALE DRIVE]
PRINT [RICHARDSON, TEXAS 75081]
.PRINTER 0
END
```

From this, one need hit only the "A" and RETURN to print out the address on an envelope or a mailing label. To print the salutation of a letter requires the simple addition of another procedure.

```
TO S
A
.PRINTER 1
PRINT []
PRINT [DEAR MR. MULLER:]
PRINT []
.PRINTER 0
END
```

Other single keystroke entries could allow a handicapped person to printout an entire letter using 20 to 30 keystrokes rather than the 200-300 that might otherwise be required.

"The Toddler's Turtle" is a single-keystroke Logo procedure developed by a Richardson teenager to help kindergarten children and the disabled focus their attention on procedural, structured thinking. On their own, at their own pace, preschoolers enjoy a structured approach to graphic design and development.

The beautiful part of Logo is that it is limited only by the imagination. And nowhere is this better demonstrated than through the eyes of a child.

There is a certain bit of magic that fills the air watching a hyperactive teenager sit for an hour and a half, programming his own game of Lunar Lander. There is a unique exceitement watching a three year old manipulate the Turtle around the screen in precisely directed patterns. There is a fascinating serenity that fills a room when a group of youngsters work together to develop a linguistic procedure to translate English into Spanish, or when a group of third grade students explore the spherical geometry concepts in the patterns of a soccer ball. It is intriguing to listen to the discussion of strategic gun emplacements as students analyze the star-shape of Fort McHenry, the home of the Star Spangled Banner as drawn by the Turtle (See Tables III through VI).

Most certainly, this form of learning defies the typical "rote learning" of years gone by. The dictionary defines it as "the use of memory, usually without intelligence." Where once this suited the educational process, the speed of technological change demands that children not only assimilate knowledge but they learn to think effectively for themselves, that they learn to use their creativity and imagination to best advantage. Margaret Mead said it quite well in her description of the information explosion. She saw that people had learned to gather, analyze, sort, and disseminate information at totally incomprehensible speeds. But these same people have failed to synthesize this new knowledge. They have no idea of how to mesh it into their lives. Teaching young people to think logically and imaginatively may well be a step in the right direction.

It is easy to believe that Einstein had Logo in mind when he said that "imagination is more powerful than knowledge." The ability to create new information through creative discovery is far more powerful and meaningful than the mere recitation of memorized facts.

Even more appropriate to Logo is another quotation of unknown origin: "he who has imagination without learning, has wings but no feet."

On the stable platform of logical and developmental thinking, of self-discovery and self-reliance, of analytical skill development and decision-making, young people are able to plant their feet so their imaginations can soar with the eagles.

It is a most exciting and brave new world they are entering--riding on the back of a Turtle.

```
TO HELP
CLEARTEXT SETCURSOR [0 8]
PR [THIS PROCEDURE CAN HELP YOU WITH THE]
PR [PARTS OF SPEECH: ARTICLES, ADJECTIVES]
PR [NOUNS, AND VERBS. CAN YOU USE EACH]
PR [CORRECTLY? TO FIND OUT, JUST FOLOWTHE]
PR [DIRECTIONS ON THE SCREEN.]
PR [GOOD LUCK! TO START, PRESS 'RETURN.]
MAKE "ANS READLIST CLEARTEXT
END

TO PICK :LIST
IF :LIST = [] OUTPUT 0
OUTPUT (PICK BUTFIRST :LIST) + 1
END

TO NUMBER :LENGTH
MAKE "N (RANDOM 3) + 1
IF AND (:N > 0) (:N < (:LENGTH + 1)) (OUTPUT :N)
IF OR (:N > 0) (:N < (:LENGTH + 1)) (OUTPUT NUMBER :LENGTH)
END

TO SELECT :N :LIST
IF :N = 1 [OUTPUT FIRST :LIST]
OUTPUT SELECT :N - 1 BUTFIRST :LIST
END

TO WORDS :X
(TYPE SELECT (NUMBER (PICK :X)) :X " )
END

TO GRAMMAR :ART :NOUNS :ADJ :VERBS
WORDS :ART
WORDS :ADJ
WORDS :NOUNS
WORDS :VERBS
WORDS :ART
WORDS :ADJ
WORDS :NOUNS
TYPE ".
PRINT []
PRINT []
WAIT 60
GRAMMAR :ART :NOUNS :ADJ :VERBS
END

TO SENTENCES
HELP
PRINT [TYPE A LIST OF ARTICLES AND THEN PRESS]
PRINT [ENTER.]
MAKE "ART READLIST
PRINT [TYPE A LIST OF NOUNS AND THEN PRESS]
PRINT [ENTER.]
MAKE "NOUNS READLIST
PRINT [TYPE A LIST OF ADJECTIVES AND THEN]
PRINT [PRESS ENTER.]
MAKE "ADJ READLIST
PRINT [TYPE A LIST OF VERBS AND THEN PRESS]
PRINT [ENTER.]
MAKE "VERBS READLIST
PRINT [NOW WATCH!] WAIT 80 CLEARTEXT
GRAMMAR :ART :NOUNS :ADJ :VERBS
END

?
```

Table I

```
THIS PROCEDURE CAN HELP YOU WITH THE
PARTS OF SPEECH: ARTICLES, ADJECTIVES
NOUNS, AND VERBS. CAN YOU USE EACH
CORRECTLY? TO FIND OUT, JUST FOLOWTHE
DIRECTIONS ON THE SCREEN.

GOOD LUCK!

TO START, PRESS "RETURN."

TYPE A LIST OF ARTICLES AND THEN PRESS RETURN.

A AN THE

TYPE A LIST OF NOUNS AND THEN PRESS RETURN.

DOG CAT HORSE BOY GIRL

TYPE A LIST OF ADJECTIVES AND THEN PRESS RETURN.

CUTE TALL SMALL FAT THIN

TYPE A LIST OF VERBS AND THEN PRESS RETURN.

SEES KISSES LIKES

NOW WATCH!
THE BIG HORSE LIKES THE SMALL CAT .

A BIG HORSE LIKES THE CUTE CAT .

AN SMALL HORSE LIKES A CUTE DOG .

AN BIG CAT SEES THE BIG HORSE .

A CUTE DOG LIKES AN CUTE DOG .

A CUTE CAT SEES A SMALL CAT .

AN BIG CAT LIKES A CUTE CAT .

AN BIG DOG KISSES A CUTE CAT .

A CUTE DOG SEES A BIG CAT .

AN BIG CAT LIKES A SMALL CAT .

THE BIG DOG KISSES AN BIG CAT .

THE SMALL CAT LIKES THE BIG CAT .

A SMALL HORSE KISSES A BIG CAT .

THE SMALL CAT SEES AN CUTE CAT .

AN SMALL HORSE SEES THE SMALL CAT .
```

Table II

```
TO INSIDE.CORNER
FORWARD 56 LEFT 72 FORWARD 20 RIGHT 72
REPEAT 2 [FORWARD 10 RIGHT 72]
FORWARD 20 LEFT 72
END

TO CORNER
FORWARD 50 LEFT 72
REPEAT 3 [FORWARD 20 RT 72]
FORWARD 20 LEFT 72
END

TO RAVELIN
REPEAT 2 [FORWARD 40 RIGHT 120]
FORWARD 7 RIGHT 60 FORWARD 25 LEFT 120 FORWARD 25 RIGHT 60 FORWARD 7
HIDETURTLE
END

TO MOVE2
PENUP BACK 10 LEFT 72 FORWARD 50 RIGHT 15 PENDOWN
END

TO INSIDE.WALL
REPEAT 5 [INSIDE.CORNER]
END

TO MOVE1
PENUP RIGHT 90 FORWARD 8 LEFT 90 BACK 3 PENDOWN
END

TO WALL
REPEAT 5 [CORNER]
END

TO MOVE
FULLSCREEN
PENUP FORWARD 20 PENDOWN
RIGHT 90
END

TO FORT.MCHENRY
MOVE
WALL
MOVE1
INSIDE.WALL
MOVE2
RAVELIN
END
```

Table III

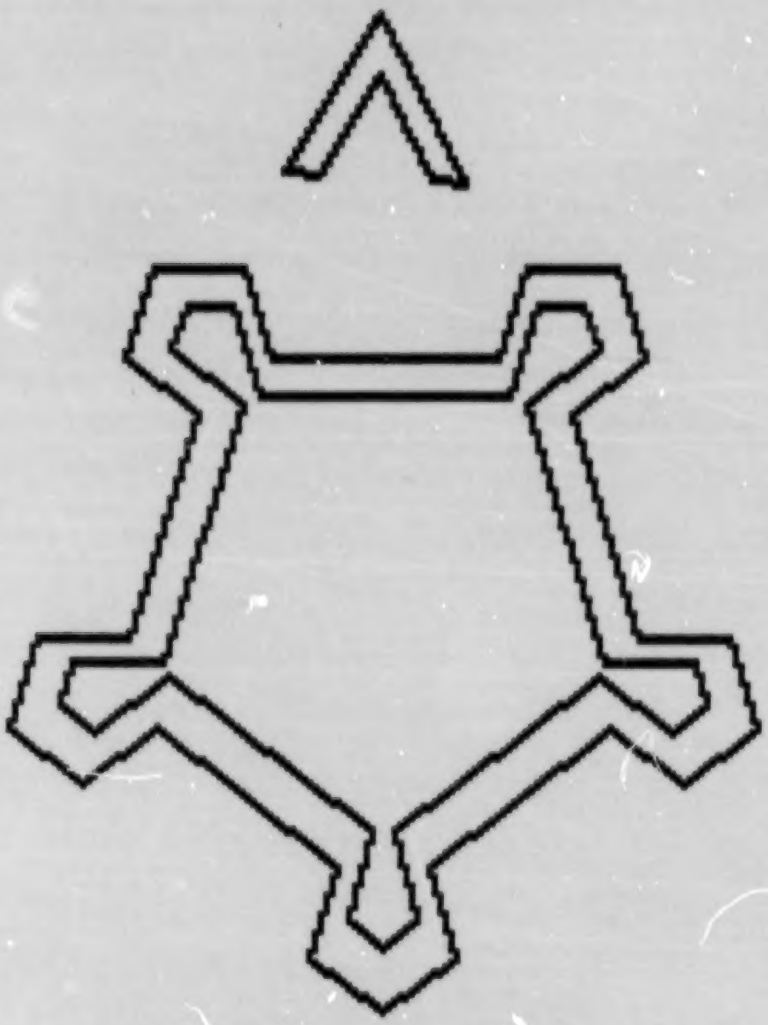


Table IV

```
?TO LTURN
LEFT 90 FORWARD 14.5 LEFT 90
END

TO RTURN
RIGHT 90 FORWARD 14.5 RIGHT 90
END

TO STRIPES
FORWARD 116 RIGHT 90 FORWARD 246
REPEAT 4 [RTURN FORWARD 130 LTURN FORWARD 130]
REPEAT 3 [RTURN FORWARD 246 LTURN FORWARD 246]
RIGHT 90 FORWARD 14.5
REPEAT 2 [RIGHT 90 FORWARD 246 RIGHT 90 FORWARD 217.5]
END

TO BOX
PENUP SETPOS [-130 -12] PENDOWN
SETHEADING 0
REPEAT 4 [FORWARD 116 RIGHT 90]
END

TO MOVE5
PENUP RIGHT 75 FORWARD 34 LEFT 75 PENDOWN
END

TO STAR :N
REPEAT 5 [FORWARD :N RIGHT 144 FORWARD :N LEFT 72]
END

TO MOVE4
LEFT 15 PENUP BACK 20 LEFT 90 FORWARD 85 RIGHT 90 PENDOWN
END

TO MOVE3
LEFT 15 PENUP BACK 20 LEFT 90 FORWARD 50 RIGHT 90 PENDOWN
END

TO ROW
RIGHT 15
REPEAT 2 [STAR 7 MOVE5]
STAR 7
END

TO STARS
REPEAT 2 [ROW MOVE3 ROW MOVE4] ROW
END

TO BEGIN
PENUP SETPOS [-115 90] PENDOWN
END

TO FLAG
BEGIN
STARS
BOX
STRIPES
END

?
```

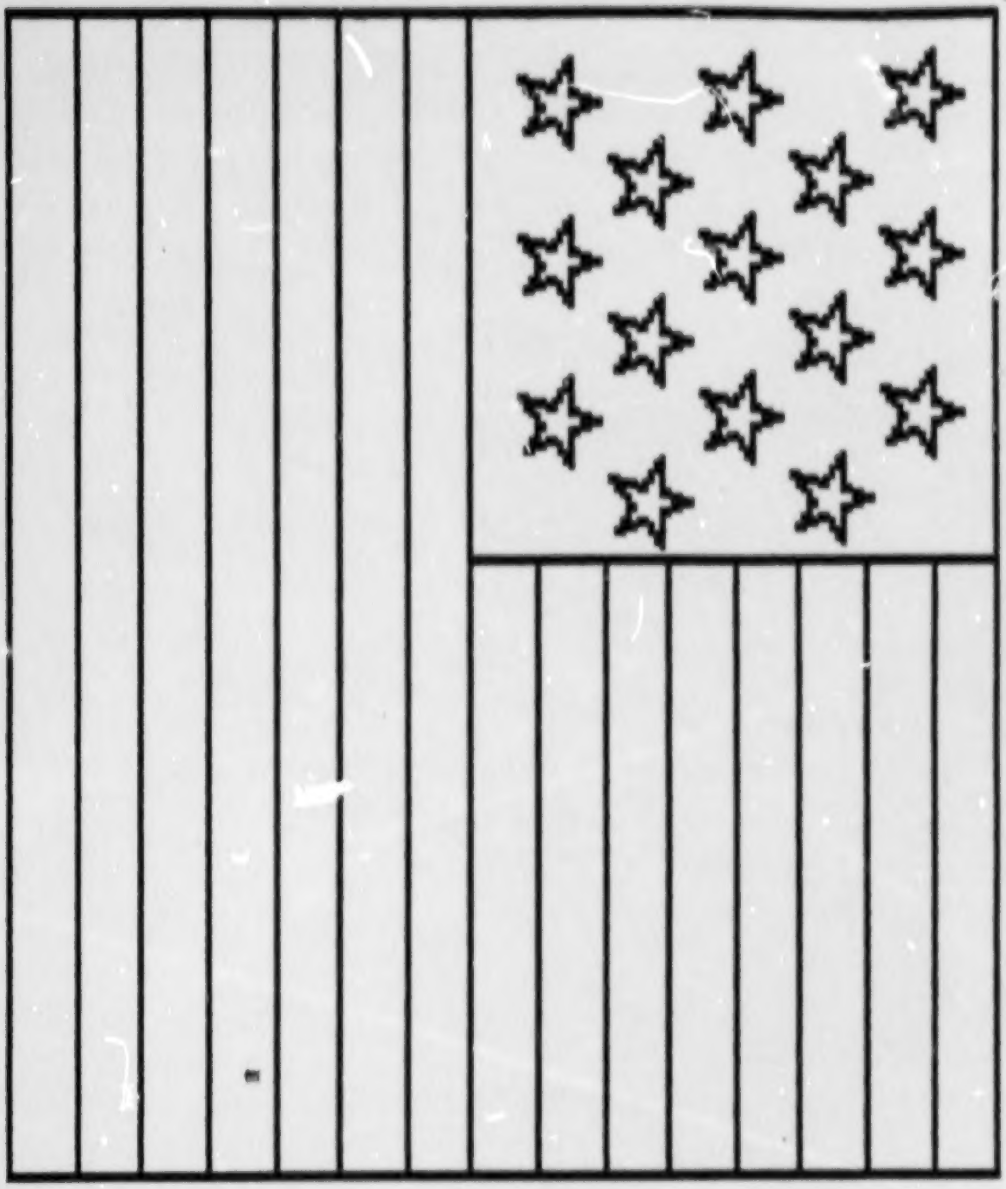


Table VI

Part II

Abstracts of Papers Presented

2.1 Education Administrators

2.1.1. One School District's Reponse to Accountability For Student Progress in Reading and Mathematics

Martha W. Combs, Oklahoma State University

In 1979, the Florida legislature enacted the Florida Primary Education Program which was designed to identify and provide an appropriate instructional program for all students who were considered to be ahead or behind developmental expectations for their particular grade level. Consequently, a record keeping and instrucational assistance system was devised by personnel in one county school system that utilized microcomputers. This system allowed the following improvements:

1. Charting student progress by class, grade, and/or school.
2. Generating individual instruction plans to aid teachers in planning an appropriate educational program.
3. Assisting teachers in the selection of instructional materials for students based upon modality preference and instructional setting.
4. Assisting teachers in analyzing their instructional effectiveness on an ongoing basis.
5. Improving feedback to parents on the progress of their child.

In this project, the microcomputer proved to be a valuable tool for instructional planning and decision making. The process was slow at first as data had to be entered. Once files were established, information was easily retrieved and useful to teachers. Communication between schools, as children moved within the school district, improved with the quality and uniformity of data avilable. Administrative personnel were able to compile data concerning K-3 students more frequently and continuously evaluate the primary grades instructional program.

The greatest benefit, along with more efficient use of the teachers' planning time, may have been the process of sensitizing teachers to individual student needs on a continuous basis. Within and between grade levels, planning and cooperation improved. Student needs superceded administrative convenience for placement and pacing of children.

2.1.2. Computing Readability by Microcomputer

Karen Connel, Stillwater Public Schools, Stillwater, Oklahoma
Susan Mills, Oklahoma State University

The readability level or reading difficulty of textbooks and other reading materials has traditionally been calculated on the basis of percent of "unfamiliar" words or number of syllables per 100-word passage and expressed as a grade level. In the past, these calculations have been performed by hand. It is not surprising, with the increasing availability of micro-computers, that more and more programs are appearing on the market. When selecting a readability program, it is advisable to consider the following factors:

1. What readability formulae are included in the program?

   It is a good idea to have more than one readability level so that comparisons may be made. Most formulae provide an estimate of reading difficulty and by obtaining several scores, such as Fry, Dale-Chall or Spache, Fog, and Flesch, a more accurate estimate of the range of readability of a passage can be obtained.

2. Is there adequate explanation of how to interpret the reading scores obtained?

   A reading score expressed in terms of grade level, such as 3.5, is relatively straight-forward and easy to understand. However, if the score is expressed as some other figure, there should be a standard scale against which to compare the readability score.

3. If a typing error occurs while inputting text, how is it corrected?

   If the entire passage must be retyped once it has been entered in order to correct a single misspelling, the process may become so time-consuming as to negate the advantage of using the computer.

4. How long does it take the program to analyze the passage of text?

   A program which requires that one line of text be entered at a time and then takes an extended amount of time to compare the passage to an "unfamiliar" word list or to count syllables again may not be time-efficient.

5. Do complete instructions for use of the program accompany it?

   Nothing is more frustrating than to be confronted by a blank screen and a blinking cursor with no idea of what to do next. The documentation of the program, along with the program itself, should provide clear, concise instructions for the user.

Two readability programs for use on the Apple computer were analyzed in order to determine how well they fulfilled the preceding requirements. "Readability Calculations" by Micro Power and Light Co. did not fulfill any of the criteria to the user's satisfaction. "Finding Readability Levels" by MECC School Utilities, however, proved to be well-constructed and highly efficient.

Resources

Readability Calculations
Micro Power & Light Co.
1200 Hillcrest Road, Suite 224
Dallas, TX 75230

Finding Readability Levels
MECC School Utilities Volume 2
Sunburst Communications
39 Washington Avenue
Pleasantville, NY 10570

2.1.3. Teachers and Computers Meeting the Challenge

Charles A. Hanna, Haskell Middle School, Broken Arrow, OK

Across the nation, teachers are being mobilized to educate the nation in computer literacy and the use of computers in the classroom. Unfortunately, most teachers fear this movement and are unprepared for various reasons. These problems add to the pressures already facing the classroom teacher.

Several options exist for computing training programs for teachers. They include:

1. self-training/independent study
2. local college coursework
3. state implemented inservice programs
4. local school district inservice

Using the local school dsitrict option, the following is a two-part approach to help serve as a guide. The first part of the presentation deals with the task of planning the program. The second details an example of a program currently being implemented at Haskell Middle School in Broken Arrow, OK.

Part I

A. Recognition of Needs

1. The present and the future
2. Trends in the business world
3. Responsibility of the school district

B. A Plan of Action

1. Task force formation
2. Mandate for the Board of Education
3. Development of expectations district-wide K-12
4. Selection of hardware
5. Development of a core staff
6. Development of a timeline

C. Implementation of the Plan

1. Training of core staff
2. Training of teaching staff
3. Implementation of district goals

D. Evaluation

1. Review
2. Suggestions for improvement

Part II

A. Description of the School Setting
B. Equipment and Facilities Available
C. Instruction and Guidance
D. A Self-Teaching Manual for Open Lab

Reference

A Nation at Risk: The Imperative for Educational Reform. National Commission on Excellence in Education, Washington, D. C.: United States Department of Education, 1983.

2.1.4. Microcomputer Use in Selected Small Oklahoma Schools

Richard Hunter, Wakita Public Schools, Wakita, OK

Wakita Public Schools surveyed superintendents of 65 randomly selected Oklahoma K-12 schools of 400 or less students. The purpose was to generate information indicating the extent and use of microcomputers in small schools as well as to answer questions regarding implementation of microcomputers into the curricula. Of the 49 responding schools, 12 had no computers while 37 had one or more. Of these 37, seven had fewer than 200 students, 15 had 201-300

students and 15 had 301-400 students. Eight of the schools first used computers in 1983, four schools had used them over four years, and the remaining 25 had used computers for one to three years.

Twenty-six schools used TRS-80 (Radio Shack), 18 used Apple, and 12 listed other brands including Franklin which is sometimes grouped with Apple (12 had more than one brand). Eleven schools had one to four computers, 14 had 5 to 8, while 12 schools had 10 to 15.

Computers were located in a variety of locations within the schools and 19 schools had computers in multiple locations. Twenty-six had them in computer labs; 12 in mathematics classes; 6 in reading labs; 6 in the library or media center; 10 in other high school classrooms; 9 in elementary classrooms; and 9 in the school office.

Computers were being used for a variety of instruction and/or administrative purposes, and 32 of the 37 listed multiple uses. Computer assisted instruction was listed as a use by 30 schools; computer literacy by 27; programming by 21; and computer managed instruction by 6. Eleven schools listed various administrative uses. The subjects listed by schools as areas of computer use (in order of frequency) were: computer literacy, math, business education, programming, reading, science, language arts, social studies, career education, art and music. Computer software had been purchased in a variety of disciplines with the largest expenditures in the areas of computer literacy, math, reading and business education. Amounts spent on software ranged from $0 to $5,500. Fifteen schools had spent in the $1,000 to $2,500 range.

Computers were being utilized at all grade levels in some schools but in all schools they were being used in grades 9-12. Fifteen of the schools used computers in grades K-3; 19 in grades 4-6; and 20 in grades 7-8.

Schools also listed the periodicals that contributed to their decision making regarding computer purchases and usage, companies they preferred to deal with, and strategies found to be most useful for teacher inservice in the use of computers.

2.1.5. Computerizing the Secondary School Curriculum: An Administrative Approach

Kara Gae Wilson, North Intermediate High School, Broken Arrow, OK

From the findings of the Second National Survey of Computer Use, it is apparent that the growth of computers in schools does not necessarily insure increased student use.

One approach to the dilemma is to revive the semester long mini-course approach to scheduling computer courses. The following illustrates this concept which combines (1) nine weeks of Developmental Reading Lab with (2) nine weeks of Computer Literacy into one course entitled "Technology Lab."

Each nine week course enrolls a maximum of 20 students. Thus, 40 students are exposed to two critical curriculum areas in the same time that would generally accommodate only 20 per subject.

The nine weeks computer literacy/lab could be combined with any subject area which compresses into nine weeks essential skills every student should know, i.e., study skills, science projects, technical writing, special math techniques such as understanding word problems, and understanding probability and statistics.

With this type of scheduling it is possible to tie computer lab to a less popular student elective course, but one which administrators and teachers feel is equally important to students in the future. This approach is designed to extend the access of computers to the greatest number of students of all ranges, while at the same time focusing on additional, perhaps even more substantial learning skills.

2.2 Elementary Teachers

2.2.1. Variables for Students Under Ten

James E. Evans, Broken Arrow Senior High School, Broken Arrow, OK

During the summer of 1983, Tulsa Junior College and Tulsa area public schools cooperated in a program to teach programming skills to students aged eight to eighteen. BASIC language and Texas Instruments TI 99/4a computers were utilized.

Comprehension of the use of variables was found to be a problem when dealing with the younger students. Several factors were identified which seemed to assist the young students in their use of variables.

The most important factor was found to be motivation. Specifically, when the students discovered that variables could be used to save typing, they became much more motivated to employ variables in their BASIC programs. The discovery that redefining variables could save typing also proved motivating and aided the students' comprehension. Finally, the discovery that variables can be used to form new values (e.g. X=Y+Y) also proved motivating.

In summary, it was the thesis of this paper that the need of the student to apply variables is the greatest factor in the student's ability to successfully use variable concepts.

2.2.2. Implementing the "Impossible": One Computer and Thirty Kids

Jo Ann McEwen, Skyline Elementary School, Stillwater, OK

Three years ago, I decided my students were being denied the technical knowledge of computers in the school, so I started taking my computer to school. My first objective was for each student to have 15 minutes hands-on experience with the computer before I started to use it to reinforce lessons through computer-assisted instruction.

During a short detailed description of turning on the computer and loading a program, students eagerly took notes. Everyone wanted to be the first one to use the computer. I solved this by dividing the time my students were actually in the room into 15 minute slots. I gave each student a number and randomly drew student's numbers to fill the slots for their individual computer time. Mornings, recess, and after school were left open. Within a day and a half, every student had experienced loading a program and running it. This system worked so well that I continued to use it: every other day, I drew the numbers and posted them. Students never had to be reminded that it was their turn to begin or end. They took care of it. Students who had to go to another class during their scheduled time found a student with which to switch times. Time before class began and during recess was reserved for students who really wanted the computer. After the first week of school, I had

students coming in at 7:30 in the morning for extra computer time. I never insisted on students using the computer after the first hands-on experience. If a student didn't want to use the computer for some reason, they found another student to take their place. The students had a free choice as to which program they wanted to run unless I wanted to use the computer as a reinforcement tool and told them which program to load.

My time involvement or responsibilities were to bring the computer each day and make sure the programs were in working condition. I soon learned to have an extra copy of each tape in case one was damaged. Since this was a personal goal of mine, the total expense of this project came out of my pocket. The programs that my students had available were programs that I wrote. But in doing this, I was able to provide the students with the kind of individualized help they needed.

I used this plan for two years in a self contained classroom and am now using it with 120 students in an open school. Each student has been exposed to the computer and can sign up for computer time during their breaks. It's not the best program, but I still have just one computer.

Teachers must decide what their goals are and how much time they want to devote to having a computer in the classroom. Some teachers will only want to know how to turn the computer or and load a program, others will want to get involved in selecting the software, and others will want to program. I believ the decision should be left up to individual teachers. Computers are constantly changing and it is impossible for teachers to know all the answers. Many students will have a greater knowledge of computers than their teachers, but is this a good reason to make our kids wait any longer?

"Let's learn together"

2.2.3. LOGO: Overview

Malcom Phelps, Oklahoma State University

LOGO is a language for learning. It is the only computer language specifically designed for education. Other languages, such as BASIC or FORTRAN, were developed for other applications and have serious flaws when applied to education. This workshop will introduce the essential elements of the LOGO language and provide an opportunity to experience LOGO's power.

One reason for LOGO's importance to educators is that LOGO is a language with "no threshold and no ceiling." Seymour Papert (1981) states that LOGO was designed such that it would be accessible to young children yet challenging for advanced programmers. LOGO is therefore effective as a language for children.

The full power of LOGO is not associated with its functions as a computer language, but rather is associated with the manner in which LOGO encourages the development of logical problem solving abilities and ways of thinking. Specifically, Papert encouraged the application of LOGO to education as a result of his five-year experience in study with Jean Piaget.

This workshop presentation has two main goals. The first goal is to introduce educators to the educational philosophy which supports the application of LOGO in the classroom, especially as LOGO is compared to other computer education applications. The second goal is to familiarize participants with some of the fundamental LOGO commands. This will be accomplished through actual experience using the language to develop geometric designs.

REFERENCES

Abelson, H. LOGO for the Apple II. Peterborough, N.H.: Byte/McGraw Hill Press, 1982.

LOGO Group D.S.R.E. Bibliography of LOGO Memos. Cambridge, Mass.: M.I.T. Press, 1980.

Moore, M. LOGO Discoveries. Palo Alto, CA.: Cretive Publications, 1983.

Papert, S. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, Inc., 1980.

Taylor, R. P. The Computer in the School: Tutor, Tool, Tutee. New York: Teachr's College Press, 1980.

Watt, D. Learning with LOGO. New York: McGraw-Hill, 1983.

2.3 Secondary Teachers

2.3.1. Using Microcomputers for Editing Business Letters and Reports

Karen Forcht, Oklahoma State University

One of the most valuable uses of microcomputers and word processors in composition instruction is in the teaching of writing revision. One of the most frustrating tasks that students of written communication encounter is that of editing. To alleviate this numerous writing revision task and thus afford the student the opportunity of viewing the end product sooner, many teachers are using the computer or a word processor as a revision tool.

A number of activities can be carried out with this method. For example:

1. After appropriate instruction on revision techniques, students can be given a prewritten text and be asked to revise it on the computer or word processor.
2. After a student has composed text, the computer can suggest that certain revisions be made.
3. After a student has composed text, the computer can search for specific kinds of errors, mark the place where the errors occur, and require the students to correct them.
4. The computer can also be used to teach the students sentence structure, combining sentences, and how to generate and arrange content.
5. The teacher could also act as editor, interacting with student writing in ways similar to those suggested above.

The automated devices make the work of revision much easier. Major and minor changes can be made without having to recopy. The changes take place right before the student's eyes and students have the opportunity of viewing their "clean" text immediately.

Computers are definitely changing the way we write today and we, as teachers, must adopt this new method and incorporate the concept into our classroom instruction in order to keep pace with the automated world our students are entering.

REFERENCES

Automated Dictionairies, Reading and writing: Chairman's report of a Conference on Education Uses of Word Processors with Dictionairies (New York, New York, June 14-15, 1979), by the National Institute of Education (DHEW), Washington, D. C., Dec. '79.

Burns, Hugh L., Jr., "Stimulating Rhetorical Invention in English Composition Through Computer-Assisted Instruction", 1979.

Collier, Richard M., "The Influence of Computer-Based Text Editors on the Revision Strategies of Inexperienced Writers," October 1981.

Cronnell, Bruce and Ann Humes, "Using Microcomputers for Composition Instruction," National Institute of Education, Washington, D. C., March, 1981.

Emif, Janet. "Writing as a Mode of Learning," College Composition and Communication, May, 1977, pp. 122-128.

"Fast Growth Expected in Speech Processing," Word Processing World, September, 1978, p. 8.

"Function and Form in the Automated Office," Datapro, November, 1978, pp. 114-117.

Flower, Linda, "Writer Based Prose: A Cognitive Basis for Problems in Writing," College English, Vol. 41, No. 4, September, 1979, pp. 19-37.

Gould, John, "Experiments on Composing Letters," Cognitive Processes in Writing, eds. Greff, L. W. and E. G. Steinberg, Hillsdale, N.J.: Lawrence Erlbaum, 1980, pp. 97-127.

Halpern, Jeanne W., "Paper Voices: How Dictation and Word Processing are Changing the Way College Graduates Write," research paper prepared at Purdue University, 1980.

Herbert, Mellanie, "You Can Do It! Modify Software for your Students' Needs," Journal of Business Education, February, 1983, pp. 178-180.

Golen, Steven, "Using a Data Processing Tool in Business Report Writing Class," Journal of Business Education, April, 1982, pp. 266-267.

Ong, Walter, Interfaces of the Word: Studies in the Evaluation of Consciousness and Culture (Ithaca: Connell University Press, 1977), p. 86.

Smelstor, Marjorie, Ed. A Guide to Teaching the Writing Process from Pre-Writing to Editing. Wisconsin University, Madison, Department of English, School of Education, University Extension, Sept. 1978.

"WPS Measures Dictation Usage," Word Processing Systems, February, 1980.

2.3.2. PASCAL OR IS THERE LIFE AFTER BASIC?

PHILIP KNIGHT, PERKINS-TRYON SCHOOLS, PERKINS, OK

If one were to survey those school districts offering a high school course in programming, the chances are that course would be taught in BASIC. However, this situation is rapidly changing. One major reason for this change is that introductory computer science courses at most universities are now being taught in PASCAL rather than FORTRAN 77 or one of its many variants. Additionally, to accommodate for these changes being made by colleges and universities, Educational Testing Service (ETS) will soon offer - May of 1984 -

advanced placement in computer science to high school students who can demonstrate proficiency in programming in PASCAL. These dramatic changes lead one to ask the question, "Why PASCAL?"

PASCAL was developed to serve as a general purpose procedure-oriented programming language rather than a number crunching or text manipulating language. Programming in PASCAL is done in units called "blocks" rather than in "spaghetti code" as is characteristic of longer programs written in BASIC. Block or modular programming facilitates ease in reading a PASCAL program because the program itself is a compilation of clear, self-documenting modules. A consequence of this type of programming is that managers are more able to control program development when multiple programmers are involved. Neither FORTRAN or BASIC provide for this degree of programming independence.

PASCAL is a compiler language. This provides the dual advantages of quicker program execution time and less memory usage. However, it would be misleading not to point out that most microcomputer generated PASCAL programs are not true N-code compiled programs but rather P-code compiled programs.

One of PASCAL's greatest strengths is the ease by which it is able to manipulate tables and lists of data. To appreciate this point one need only compare the time spent doing these activities when programming in BASIC and then programming in PASCAL.

Though it appears that PASCAL is without fault, this is not the case. One major disadvantage of PASCAL is the difficulty in debugging a program. Even with compiler assistance, there is no comparison in ease of debugging an interpreter BASIC program. To believe is to experience!

PASCAL is considered a more verbose language than, for example, BASIC. What might take 15 lines in BASIC might take 50 lines in PASCAL. PASCAL's advantage over BASIC is not realized until programs are large in size.

Other disadvantages of PASCAL become apparent when one better understands its syntax and structure. For example, the awkward manner in which input errors are handled, limited I/O controls, consequences of using mixed data types, etc. Even the cost of the software and/or firmware for PASCAL can be prohibitive. Yet, overall, the advantages of PASCAL outweigh the disadvantages. Ask a PASCAL programmer.

RECOMMENDED READING

Apple. The Apple PASCAL Language Reference Manual. Cupertino, CA. Apple Computer, Inc. 1980.

Apple. The Apple PASCAL Operating System Reference Manual. Cupertino, CA. Apple Computer, Inc. 1980.

Bowles, K. L. Microcomputer Problem Solving Using PASCAL. New York, N.Y.: Springer-Verlag. 1977.

Clark, R. & Koehler, S. The UCSD PASCAL Handbook. Englewood Cliffs, N.J.: Prentice-Hall, Inc. 1982.

Jensen, K. & Wirth, N. PASCAL User Manual and Report. 2nd ed. New York, N.Y.: Springer-Verlag. 1978.

Luehrmann, A. & Peckham, H. Apple PASCAL: A Hands on Approach. New York, N.Y.: McGraw-Hill, Inc. 1981.

MacCallum, I. PASCAL For The Apple. Englewood Cliffs, N.J.: Prentice-Hall. 1983.

Richards, J. PASCAL. New York, N.Y.: Academic Press. 1982.

Seiter, C. & Weiss, R. PASCAL for BASIC Programmers. Reading, Mass: Addison-Wesley. 1983.

2.3.3. Building the High School Computer Curriculum

Tony Lutman, Broken Arrow Senior High School, Broken Arrow, OK

This session focused on the important considerations necessary in designing and planning an instructional computing plan for a high school. Those considerations discussed were:

(a) Hardware considerations
(b) Selecting a location and planning the facility
(c) Number and ability level of students using the computer
(d) Type of coursework to be taught using the computer - programming vs. computer literacy
(e) Types of programming languages to be taught
(f) Inservice necessary for teachers using computers
(g) Incorporation of computers in other curriculum areas
(h) Implementation and access of computers
(i) Future considerations
(j) School and parent involvement

Under the direction of the mathematics department, three courses were developed for the 11th and 12th grade. They include "General Computer Programming," "Math/Science Programming," and "Advanced Math/Science Programming." The first course is intended for liberal arts students with minimal math background (2.5 GPA minimum, "C" or above in previous math courses). Students are not allowed to take further programming courses. The second course requires the students to have completed or be concurrently enrolled in the third year of mathematics, "B" or above in all math courses. The third course requires completion of the second course and teacher approval. Course outlines and textbook references for each course are listed below.

I. General Computer Programming
Texts: Jones. I Speak Basic to My TRS-80. Hayden.
Clark. Structured BASIC. South-western.
Shelly. Introduction to BASIC Programming. Anaheim.

A. History and Terminology
B. Flowcharting
C. Elementary Programs
D. Graphics
E. Scripsit
F. Simple Arrays
G. Data Storage and Retrieval
H. Random Function

II. Math/Science Programming
Text: Golden, N. Computer Programming in the BASIC Language. Harcourt, Brace and Jovanovich (Second Edition).

A. Flowcharting
B. Elementary Programs
C. Controlling Output: If...Then
D. Loops

III. Advanced Math/Science Programming
Text: (same as for the second course)

A. Loops
B. Subscripted Variables
C. Matrices
D. Additional Features of BASIC
E. Individual Project

2.3.4. Teaching of Arrays with Applications

Paul Ohme, Northeast Louisiana University
Lejeane Thomas, Louisiana Tech University

For most nonquantitative professions and for many quantitative problems, the chief purpose of the computer is to store large quantities of data in order to perform on it various manipulations. This makes the topic of arrays a major focal point in any programming course, particularly for those outside of mathematical sciences. However, leading a novice programmer to the skill of working with arrays is difficult. This presentation began with the idea of extending the single dwelling for data concept (non-dimensional variables) to the idea of apartment complexes (dimensioned variables) and the need for advanced zoning clearances or reservation (DIM) for these complexes. Various techniques for placing tenants into an apartment are described. A series of problems are introduced, each of whose solution will be used to build understanding of arrays and the method of approaching the programming of an array problem.

2.3.5. Teaching Programming: An Initial Experience in Problem Solving

Thomas R. Rahlfs, Kansas State University, Manhattan, KS

Confusion has existed between 'programming per se' and the learning of a high level programming language. What is meant by programming for this paper is the entire series of steps involved in solving a problem on a computer. Too often the word programming has become synonymous with the word "coding" which is simply the process of writing statements in the syntax of a particular programming language.

Within many computer classrooms, programming has become nothing more than a simple coding activity. The preparation involving defining the problem, deciding and outlining the solution have not been included in the programming instructional activities. Instruction limiting the scope of instruction in programming to learning code syntax has been doing the student a disservice. Wexelblat (1981) quoted:

> "If language influences thought patterns, then we should teach programming before we teach the language. Otherwise biases become ingrained."

The following model was outlined suggesting steps for teaching programming.

I. Defining the problem - clarification of exact specifications desired.

   A. Three categories of questions concerning problem specifications
      1. input specifications
      2. output specifications
      3. special processing

II. Outlining the solution - designating specific and interralated program tasks

III. Selecting and representing algorithms

   A. Description of algorithm independent of any specific language
      1. Statement of algorithm indicating variable assignment, processing, branching and looping
      2. Flowchart the program

IV. Coding - choice of language

V. Debugging - location of problems and correction of errors

VI. Testing and Validation - verifying results are correct in all cases

VII. Documenting - part of a complete program

A. Technical documentation for the programmer using remark statements within the program
B. User level documentation including operating instructions both machine and software

VIII. Program Maintenance - necessary updating to satisfy new needs of program

This sequence of problem solving has had a beneficial effect upon students involved in programming courses. The author has also suggested this process may have a positive effect upon student problem solving skills in other areas.

REFERENCES

Schneider, G. Michael; Weingart, Steven W.; and Perlman, David M. An introduction to programming and problem solving with PASCAL. (Second Edition) John Wiley & Sons, New York, 1978.

Wexelblat, Richard L., The consequences of one's first programming language. Software-Practice and Experience, 1981, 11, 733-740.

2.4 Special Educators

2.4.1. Using the Computer with Gifted and Talented

Kay Sather Bull, Oklahoma State University

Six facets of microcomputer usage were examined in relation to gifted education programs. They were:

(1) Microcomputer's role in gifted and talented education
(2) Curricular differentiation and the microcomputer
(3) Various types of software systems (e.g., CAI, CMI, problem solving, and programming) and their modification for the gifted and talented
(4) Benefits and changes expected for gifted and talented students
(5) Key points in selecting software for the gifted and talented
(6) Appropriateness of computers for gifted and talented education and increasing their effectiveness

The focus should be on education and not computers when considering the role of microcomputers in gifted education. Education needs to set some specific goals for microcomputer incorporation that concentrate on instructional experiences different from drill and practice.

Computer based experiences should be designed based on the five legitimate methods of differentiating curriculum for the gifted. In some cases, students are exposed to a variety of software programs that are unrelated to the scope and sequence curriculum model designed for a student's gifted area. These experiences, called horizontal enrichment, are not appropriate for the gifted and talented student. CAI and programming instruction should be developed using methods emphasizing acceleration, augmentation, expansion (to higher thinking skills), research, independent study, and parallel tracking.

Several ways of using the computer in gifted education include CAI, CMI, programming, and discovery experiences. Two types of CAI packages, representation and simulation, are the most appropriate for the gifted and talented. CMI packages provide more flexibility and independence for teacher and student. Programming is not meant for all gifted and talented students. The main goal of this type of computer-based instruction is the encouragement of the student's thinking processes. Van Horn (1982), Grady (1983), Dwyer & Critchfield (1981) list several suggestions outlining content areas and techniques that can be used to teach programming to the gifted and talented. Computer based discovery experiences (especially those in young children) allow the children to be in control and enhance concept formation of things that would not normally be learned until much later in life (Papert, 1981).

There are two kinds of classroom applications that should be examined when designing the computing environment. The first is the computing classroom where students learn to program. The second kind of classroom setting is the regular or pullout classroom where the gifted students will work with microcomputers as an instructional, programming, or discovery tool. Research has not produced evidence indicating the ideal number of students per machine. Experimental evidence does indicate students may be excluded if there is not much access.

Evaluation forms for the selection of software packages have not been designed for gifted education. It is recommended that the educator design a form appropriate to the particular gifted and talented program giving special consideration to such things as target population, necessary prerequisite skills, CMI features, reading level appropriateness for gifted and talented, advance organizers within the instruction, and many more.

Research has indicated benefits and expected changes with various types of computer based learning experiences for the gifted and talented student. Computers as presentation devices are appropriate when they can provide experiences that cannot otherwise be obtained by the gifted student. Computers as management devices are more than appropriate for the gifted and talented to provide both curricular flexibility and goal differentiation. Programming is appropriate primarily because it effects the thinking of students. Discovery approaches are also appropriate because they help the child become an epistemologist rather than a pawn manipulated by the machine.

A complete list of references for the research mentioned as well as a complete list of software evaluation considerations for the gifted and talented may be obtained from the author.

References

Dwyer, T. A., & Critchfield, M. (1981). Multi-computer systems for the support of inventive learning. In P. R. Smith (Ed.) Computer assisted learning, Oxford, UK: Pergamon.

Grady, M. T. (1983). The high school computer science curriculum. In M.T. Grady and J. D. Gawronski (Eds.) Computers in curriculum and instruction, Alexandria, VA: ASCD.

Papert, S. (1981). Computers and computer cultures. In J. N. Nazzaro (Ed.) Computer connections for gifted children and youth. Reston, VA: CEC

Van Horn, R. (1982). Teaching your children to program in basic. AEDS Convention Proceedings, 20, 28-31.

2.4.2. Computers in Classroom: Teachers of the Gifted

George Ann Guse, Broken Arrow Public Schools, Broken Arrow, OK

The workshop focused on an overview of the major concerns in teaching gifted students (9-12). General principles of learning were discussed. They included: (a) Learning is an experience activated by and occurring within the learner; (b) Learning is the discovery of personal meaning and relevancy; (c) Learning is sometimes a painful process; (d) Learning is highly unique and individual; (e) Learning has its richest resource in the learner's self; (f) Learning is both an emotional and an intellectual process; (g) Learning is a cooperative and collaborative process; and (h) Learning is an evolutionary process.

Software was examined in relation to these concerns and principles for the gifted students in secondary education. Selections reviewed included:

a. OKLAHOMA MICRO-VIEW. (Oklahoma State Department of Vocational and Technical Education)

A program designed to help the student prepare a profile of his/her interest and career goals.

b. GEOLOGY SEARCH. (Webster Division, McGraw Hill Book Company)

A simulation program involving students as natural resource explorers engaged in the search for oil.

c. SCRIPSIT (Radio Shack)

A word processing system that allows documents to be typed, edited, and printed.

Implementation techniques and application activities for software reviewed were discussed for the gifted classroom.

References

Kepner, H. (editor) Computers in the Classroom. Washington, D. C.: National Education Association, 1982.

Clay, K. (editor) Microcomputers in Education: A Handbook of Resources. Phoenix, AR: Oryx Press, 1982.

2.4.3. Surviving Skills for the Teacher and Student in Mainstreaming

Sandra Haven, Lamar University, Beaumont, TX

This paper will focus on the steps required in the decision making process when incorporating microcomputers into the school system. Emphasis will be given to the special educator and students with special needs. Both the evaluation of software and hardware will be considered. In conclusion, research findings concerning special education and microcomputers will be presented.

Steps necessary for the successful implementation of an instructional computing program are as follows. (1.a) Complete a needs assessment; and (b) Establish a committee of interested faculty to complete the needs assessment (The committee should address funding, hardware, software, and instructional applications). (2) Inservice education should be provided for the entire faculty. The goal of the inservice program is to make all instructors computer literate. (3.a) Consider funding resources and funding needs; and (b) Allot sufficient funds to support both hardware and software; don't support one to the detriment of the other.

Applications of computers to all classes of handicapped students have been found. Though supporting research is sparse, numerous successful teaching techniques for handicapped students have been determined.

References

Blaszczyk, Henry J. "Computing and the Handicapped: The Challenge in Education," Computer, January, 1981, 15.

Budoff, Milton and Leah Hutten. "Microcomputers in Special Education: Promises and Pitfalls," Exceptional Children, Vol. 49, Number 2, October, 1982, 120.

Chaffee, M. "Elementary and Junior High School Teachers Considering Use of Microcomputers," Classroom Computer News, 1982, 76.

Hofmeister, Alan M. "Microcomputers in Perspective," Exceptional Children, Vol. 49, Number 2, October, 1982, 120.

Taber, Florence M. Microcomputers in Special Education, Reston, VA: The Council for Exceptional Children, 1983.

2.4.4. Learning Disabled Elementary Students and CAI

KimberLee Powell-Rahlfs, Kansas State University, Manhattan, KS

Although the term "learning disability" is often used to describe populations for both academic and research purposes, educators are not in total agreement with the criteria used to define these populations. The absence of an operational definition has further complicated the issue of research. Since researchers define populations differently, it is difficult to generalize the findings of one study to others. A more reliable and consistent classification system is needed in order to alleviate this problem.

The lack of agreement of the characteristics of the LD child has been one reason for the difficulty in formulating an appropriate definition. However, children with learning disabilities are often characterized as having poor memory, poor organization, and a poor attention span.

One area in which LD children have particular difficulty is mathematics. The child may have problems with perception, expressive and receptive language abilities, attention, and spatial ability which may hinder learning of mathematic skills. Research indicates that CAI can provide many elements of instruction necessary for the learning disabled student. The microcomputer can provide the necessary repetition that teachers often cannot.

CAI has been found to:

1. be at least as effective as traditional instruction. In some cases, CAI has been found to be more effective than traditional instruction.
2. be an effective means of increasing retention.
3. decrease the amount of time necessary for students to complete a task.

4. be more effective with low ability students than average or high ability students.
5. be equally as effective as other non-traditional modes of teaching.
6. increase math achievement scores as measured by standardized achievement tests.

This presentation described research which had studied the effects of utilizing two forms of CAI (drill and practice and instructional games) to increase scores on a timed multiplication test. The test included 100 single-digit multiplication facts. This has indicated the effectiveness of these two forms of CAI when used with learning disabled elementary students.

In summary, this presentation will included the problem of identifying LD populations for research purposes, the benefits of using CAI in instruction of LD students, and the use of CAI in increasing recall of basic multiplication facts with LD students.

Refer[illegible]es

Edwards, J.; Norton, S.; Taylor.; Weiss, M.; and Dusseldorp, R. How effective is CAI research? A review of the research. Educational Leadership, 1975, 33, (2), 147-153.

Geren, K. Complete Special Education Handbook. NY: Parker Publishing Company, Inc., 1979.

Lehtinen-Rogen, L. E., How do we teach him? In H. Goldstein (Senior author), Readings in Psychology of Exceptional Children. Cinn: Special Learning Corporation, 1978.

References

Edwards, J.; Norton, S.; Taylor.; Weiss, M.; and Dusseldorp, R. How effective is CAI research? A review of the research. Educational Leadership, 1975, 33, (2), 147-153.

Geren, K. Complete Special Education Handbook. NY: Parker Publishing Company, Inc., 1979.

Lehtinen-Rogen, L. E., How do we teach him? In H. Goldstein (Senior author), Readings in Psychology of Exceptional Children. Cinn: Speical Learning Corporation, 1978.

2.4.5. Evaluating Software for Special Education

Gay C. Neuberger, Oklahoma State University

When evaluating for special education uses, software may be divided into three categories: administration, assessment, and instruction.

(1) Under PL 94-142, a good deal of administrative activity is required of teachers and administrators. Microcomputers can maintain up-to-date, orderly, retrievable records and help organize complex schedules. Some

software publishers have become aware of special education administrative needs. However, published software must be carefully evaluated to be sure it meets specific district/building requirements.

(2) Assessment involves more than obtaining a total score on a test. Also relevant are: relationship of task requirements to task performance, item analysis of correct/incorrect responses, and observed test-taking behaviors. Much routine assessment may be efficiently handled by the microcomputer. However, the teacher must be cautious that using computerized assessment techniques does not result in important data becoming lost which are required for properly informed decision-making.

(3) It is in the area of instruction where the real revolution is expected. Three factors need to be considered when evaluating instructional software: instructional content, educational adequacy, and technical adequacy. As important as educational and curricular concerns are when selecting software for special education classrooms, it is most efficient to look first at the technical adequacy of the package vis-a-vis particular student abilities and needs. Even software designed specifically for special education applications may not meet unique student needs. Software packages have their own unique characteristics which must be individually evaluated.

2.5 Postsecondary Educators

2.5.1. Mega Enthusiasm with Micros

Brenda Cantwell, Westark Community College, Fort Smith, Arkansas

QUESTIONS TO ANSWER BEFORE SELECTING SOFTWARE

1. How much instructional time is required by the teacher?
2. Are the contents of the program on the grade level you need?
3. Does the program meet the need of the majority of your students or will it be used for individual tutoring?
4. How much time does the program require to run?
5. Does the program have music or sound effects and will it be a distraction to the other students while being used?
6. Is the graphics in a form that encourages the student to incorrectly answer the questions?
7. Is a back-up copy provided by the company or are you allowed to make a back-up copy?
8. Does the program provide a summary for each student?
9. Does the program run without errors?
10. Does the program provide different levels of instruction?
11. Is there positive reinforcement in the program?
12. Can you justify the price of the program?
13. Who is going to be paying for the software?
14. Will the format of the program continue to interest the students after working on it for a few times or will they soon tire of it?
15. Is the format presented in a clear uncluttered form?
16. Does the program provide a stopping point for the student or does the student have to work all the way through the program?
17. How easy can the program be loaded?
18. Does the program have the same DOS as your computer?
19. Can the program save you time and provide adequate instructions?
20. Does the program have adequate documentation?

2.5.2. Microcomputers for Teachers: A Junior College Course

Joyce S. Friske, Oklahoma State University

"Microcomputers for Teachers" is an introductory course in instructional computing for educators. Originating at a two-year college, this three-hour course was designed to meet the computing needs of teachers K-12 in all subject areas. A five day summer workshop was planned, however, the author suggests an 8-10 day format be followed to provide maximal learning opportunities for the students. A team teaching approach was used for preparation and instruction, and was very successful for the short-term workshop.

The content of the course was comprised of five major content areas including microcomputer education topics, programming topics, issues in educational computing, development of an educational computing plan, and various demonstrations. The instruction was designed to integrate several concepts from each area with corresponding activities to reinforce the distinct yet interrelated nature of the concept areas. The activities included discussions, lectures, small group/individual projects, on-line and off-line microcomputer activities.

The major content areas with sub-topics are listed below.

1. Microcomputer Education Topics
   (a) computers in education
   (b) software - description, evaluation, and selection
   (c) hardware - description, evaluation, and selection
   (d) peripheral demonstrations - hardware with accompanying software

2. Programming Topics
   (a) microcomputer operation
   (b) system commands
   (c) Programming commands
   (d) special features

3. Issues in Educational Computing
   (a) what is computer literacy
   (b) teacher certification
   (c) K-12 computing curriculum for all types of students
   (d) research and learning
   (e) impact of technology on education

4. Educational Computing Plan
   (a) district and school integration plan
   (b) classroom computing plan

5. Demonstrations
   (a) word processing
   (b) authoring systems
   (c) music systems
   (d) robotics

Resources including texts, inservice training packages, readings from books and journals, audio visual presentations, films, video tapes, and microcomputer software were implemented to support the content and broaden the experiences of the educator. An extensive list of references can be obtained from the author.

References

Poirot, J. Computers and Education. Sterling Swift, 1980.

Poirot, J. Microcomputer Systems and Apple Basic. Sterling Swift, 1980.

Poirot, J. & Billings, K. "Microcomputers in Education," Scholastic Book Services, 1982. (In-Service Training Program)

Metzcus, R., Miller, S., & Frucci, J. "The PDK Guide: An Introduction to Microcomputer Literacy for Educators," Phi Delta Kappa, 1983. (In-Service Training Pogram)

Thompson, C. & Johnson, D. "Introduction to Computers and Educational Computing," Random House Microcomputer Courseware, 1983. (In-Service Training Program)

2.5.3. Preparing Instructional Still Visuals Using a Microcomputer

Elton Li and Daryll Ray, Oklahoma State University

The microcomputer graphics screen is flexible, effective and economical media for instructional still visuals. Displays on the graphics screen can be stored on diskettes and subsequently presented on a big screen T.V. Programs are available that simulate the opertion of a slide projector. This method of presenting instructional still visuals offers much of the advantages of photographic slides while avoiding the disadvantages such as processing delays. If more conventional methods of presentation are sought, the photographic image of the graphics screen can be made and processed into slides. With the proper hardware connected, a hardcopy of the graphics screen can be obtained from which overhead transparencies can be made.

The suitable software to produce these graphic images for use as instructional still visuals must contain sufficient features, and more importantly be user-friendly enough for instructors having limited experience with the microcomputer. The "Slide Easy" program developed at the Department of Agricultural Economics, OSU, is an example of a program designed to simplify the process of making instructional still visuals using a microcomputer. Written for the Apple II+/IIe, "Slide Easy" allows the use of a variety of background colors, text colors, text styles, and text sizes. The program is menu-driven with instructions embedded--only casual microcomputer knowledge is required. The ease of operation of this program allows both the content and format of still visuals to be locally produced by the instructor with no dependency on skilled staff, and with preparation time only minimally beyond that required in designing the contents.

References

Anderson, R. H. (1983). Selecting and developing media for instruction. New York: Van Nostrand Reinhold.

Aldrich, R., & Aldrich, D. (1980). Higher Text: A higher resolution text generator (Computer progrm and Manual). Call-A.P.P.L.E.

Knapper, C. K. (1982) Information technology and instruction. In B. Sheehan (Ed.), New directions for institution research: Information technology - innovations and applications, no. 35., San Francisco: Jossey-Bass.

Moulton, P., & Colfin, R. (1983). Personal computers, answering users needs. New York: American Management Association.

2.5.4. Bringing Microcomputers to the Faculty, Staff, and Students of a Small University

Bill W. Oldham, Harding University, Searcy, ARK

Incorporating a computer literacy dimension into the life of a small university is an interesting challenge. Methods used at Harding University in meeting this challenge is the subject of this paper.

Two major decisions were made. Computer literacy training would be made available to the education faculty and staff on a full time basis. Computer literacy modules would be inserted into existing required courses to provide students with computer training.

A Computer Literacy workshop was made available to the faculty and staff during the summer of 1983. The design, support, and execution of the workshop resulted in a highly successful experience. Subsequent short term courses are being offered on special topics.

A required liberal arts course was pinpointed to receive a major module on computers. This course will meet the needs of the students of liberal arts and students pursuing a secondary teaching program.

The second of the two required courses in mathematics will have a major module on computers. This course meets the needs of students pursuing an elementary education program.

The subcommittee of the Academic Computing Committee, aided by grass roots inputs, has attempted to give unified assistance and direction to the effort of bringing microcomputers to the faculty, staff, and students of a small university.

2.5.5. Using a Time-Sharing Computer System in the Classroom: Large Screen Projection Television as an Interactive Teaching Aid

Charles D. Rorie, Dyersburg State Community College, Dyersburg, TN

At DSCC, concern about instructor difficulties in the teaching of computer languages led to the search for a teacher aid better than the traditional chalkboard. Observation of data processing instructors struggling to demonstrate programs to large classes made it obvious that some kind of interactive display system was badly needed.

After considerable trial and error, DSCC assembled a satisfactory system that is now proving very effective in classroom use. DSCC has susccessfully integrated the campus time sharing computer system into the classroom by interfacing the computer with a large screen projection television under interactive teacher control. Some of the uses identified for this system area:

1. Computer programs may be typed in by the instructor at his class console and simultaneously be projected to a class of 50 or more students who may see the program immediately executed.
2. Instructor notes can be stored in text form on the system and be recalled at will as needed by the instructor.
3. By using a MODEM, an instructor can call up Compuserve, the Source, or any other remote data base and display the information instantly for a large class of students.

Numerous other uses of this system have been developed, and DSCC will be pleased to share their experience to assist other colleges in the development of their own teacher interactive video instruction systems.

2.5.6. Computer-Assisted Instruction in Three Sensory Modes for Foreign Language Learning

Harry Wohlert, Oklahoma State University

Learning a language, native or foreign, demands a repetitive sensory input of all language elements to facilitate long-term memory and subconscious mastery which allows a person to communicate with conscious language analysis.[1] It has also been shown that combining various sensory modes (seeing, hearing, speaking) is most successful for language learning.[2]

While traditional language laboratories were able to present such drill and practice, it was never truly interactive nor was there any of the necessary feedback for optimal learning.

The microcomputer, a cassette player with a cassette control device (CCD), and a voice recognition unit constitute a learning system with which a student can learn interactively in all three sensory modes.

This system--an Apple II (48K), a CCD (Hartley, Inc.), a standard cassette player, and a voice recognition unit (Vet-1), (Scott Instruments)--was

successfully used for learning German strong verbs in the language laboratory at Oklahoma State University.

When a student signs on, he or she selects the desired verb lesson from a menu shown on the screen of the Apple II. The lesson is then presented first in a tutorial phase to be followed by a review phase (test).

The tutorial begins with a visual presentation of the verbs on the screen and a simultaneous pronunciation from the computer-controlled cassette player. Prompted to press the space bar, the student turns the microphone on and at the same time the cassette player off. He or she must now repeat the German verb or sentence through a microphone into the voice recognition unit which is, like the cassette player, connected to the Apple II.

If the student's verbal input matches the correct statement in context and pronunciation, the computer will advance to the next item in the lesson. The correct statement, a "template", is a native pronunciation read into the computer from the lesson disk. If the verbal input does not fall into the limits of the template, the student is given one or more tries.

After the tutorial phase, the review phase will test the student about the material just studied. All items are not presented in English only, and the student is prompted to say the correct German. Each item is limited to 40 characters and a three-second duration. The cassette player is inoperative during the review phase.

A pilot study conducted by the author tested the effectiveness of voice recognition as well as student attitudes toward this new technology. The results seem to confirm that this technology can be successfully used in foreign language learning.

REFERENCES

[1]Phillip M. Tell, "The Role of Certain Acoustic and Semantic Factors at Short and Long Retention Intervals," Journal of Verbal Learning and Verbal Behavior, II (1972), pp. 455-464.

[2]Will Wagers and Karin Horn, The VBLS™ Voice-Based Learning System (Preliminary Manual), Denton, Texas: Scott Instruments, 1982, pp. 7-9.

2.6 Teacher Educators

2.6.1. Computer Science Needs in Oklahoma: Demands Exceed Supply

Richard Alexander, Hal Jenkins, Charles White, Oklahoma Baptist University

Computer science as a part of the educational curricula is beginning to permeate our educational system. Learners of all ages are becoming aware of the need for computer literacy. The question of concern, however, is directed toward the shortage of qualified public school teachers available to teach computer science courses.

With this in mind, a research project was undertaken to better understand the existing situations, the needs, and the attitudes of public school administrators relative to the teaching of computer science in the school systems of Oklahoma. One of the underlying questions of concern dealt with the determination of need for a computer science, teacher education certificate program.

A questionnaire was designed and mailed to the superintendents of the 459 independent school districts of the state of Oklahoma. There were 201 usable questionnaires returned, which is well within the statistically acceptable limits for a survey of this type.

The research confirmed the hypothesis that there is a strong and growing market in the state of Oklahoma for techers trained to teach the skills of computing. Most school districts plan to have some level of computer science instruction in their curricula within the next five years. In-service training will meet some of the needs, but there is a place for teachers who have been precisely trained to be specialists in the computer science teaching area.

On the whole, it appears that educators at every level in Oklahoma schools have already begun to prepare for the future in computer education. Schools, colleges, and departments of teacher education must begin now to prepare the teachers who will teach future generations how to use and understand computers. We are behind already!

REFERENCES

Christie, Betti J. and Dan Dolan, "Montana's Task Force on Computer Education," Technological Horizons in Education Journal, Vol. 10, No. 2, November 1982, pp. 91-92.

Lidtke, Doris Keefe, "Implementing a Program to Train (Retrain) Secondary Teachers: A Cooperative Effort Between Education and Computer Science at Towson State University," Computer Education for Colleges of Education, January 1983, pp. 78-83.

National Center for Education Statistics, "Student Use of Computers in Schools." Washington, D. C., March, 1981.

Shane, Harold G., "The Silicon Age and Education," Phi Delta Kappan, Volume 63, No. 5, January 1982, pp. 303-308.

2.6.2. Introduction to Computer Literacy on a Networking System

James Bowen, Malcom Phelps, and Susan Roth, Oklahoma State University

This presentation included both a discussion of the SWI Inc. AROS II network system which allows the Media and Technology Instruction Laboratory at Oklahoma State University to connect twenty Apple IIe computers to a central computer and a 5 megabyte hard disk and a demonstration of the operation of this system. Some of the advantages of this system include the elimination of floppy disks by allowing all students to save their own programs on a single hard disk at the central computer, and by storing educational programs on the same hard disk which each of the students at the remote computers have access to. Educational programs do not have to be loaded separately at each remote computer because each remote can view the stored programs separately or as a group. Most network systems have a special feature allowing the instructor at the central computer to "feed" a single educational program stored on the hard disk to all remote computers simultaneously.

Each student is assigned a password which allows privacy by refusing access to other users to any programs stored under that password. However, the instructor at the central computer has access to all the "public" and "private" programs stored in the system which allows easy access in grading or correcting the programs.

One disadvantage in eliminating the individual disk drives is that students cannot easily review and evaluate educational programs on floppy disks. Therefore we feel that an improvement to the system could be made by connecting floppy disk drives to at least several of the remote computers. These computers could then be easily connected to the network system or removed from the network in order to use individual floppy disk drives to view commercial software.

Overall, the system has proved very efficient and easy to work with!

2.6.3. The Role of Microcomputers in a Preservice Elementary Mathematics Methods Course: Implications for Teacher Education

Raymond Brie, Texas Christian University, Fort Worth, TX

This session began with a review of the literature related to goal three of An Agenda for Action which asks that all students have access to calculators and computers throughout their school mathematics program. This was followed by a description of a model for integrating microcomputers into an elementary mathematics methods course, including the course goals, assignments and evaluations related to microcomputers. The class is divided into four components: lecture, discussion, laboratory, and outside work. At least four class lectures are devoted specifically to teching for computer competency in mathematics education. They focus on introducing BASIC and LOGO languages, establishing criteria for reviewing and evaluating software, operating the

hardware, and discussing applications of computers in learning mathematics. Laboratory time is devoted to using various materials with computers and the outside work also requires computer applications. Specific lab experiences include 1) evaluating hardware and software, 2) programming in BASIC or LOGO and 3) programming with an authoring system.

The course competencies are:

The student will:

state ways that the microcomputer may be utilized in an elementary school mathematics program;

describe the strengths/weaknesses of software and microcomputer utilization with elementary school children and mathematics;

describe several approaches to mathematics instruction (textbook, activity, microcomputers, integrated, etc.) and defend at least one;

write or edit a computer program that promotes mathematics thinking (in BASIC or LOGO) which could be used with elementary school children.

The session ended with a discussion of the curriculum implications for teacher education in areas other than mathematics.

REFERENCES

Anandam, K. and Kelly, J. T. "Teaching and technology: Closing the Gap." Technological Horizons in Education Journal, 1982, 10 (2), 84-90.

Anderson, C. A. "Computer Literacy: Changes for Teacher Education." Journal of Teacher Education, 1983, 34 (5), 6-9.

Evans, C. The MicroMillennium. New York: Viking Press, 1979.

Friedman, D. The impact of educational computing on teacher education. Journal of Teacher Education, 1983, 34 (5), 14-18.

Foell, N. A. A new concern for teacher educators: Computer literacy. Journal of Teacher Education, 1983, 34 (5), 19-22.

Griffin, W. H. Can technology have any significant impact on education? Technological Horizons in Education Journal, 1983, 11 (3), 96-99.

Heck, W., Johnson, J. and Kanskey, R. Guidelines for evaluating computerized instructional materials. Reston, Virginia: National Council of Teachers of Mathematics, 1981.

Hunter, E. What's the low-down on high tech?: Some questions for teacher educators. Journal of Teacher Education, 1983, 34 (5), 26-28.

Judd, W. A teacher's place in the computer curriculum. Phi Delta Kappan, 1983, 65 (2), 120-122.

Mandell, A. Computer literacy and science education. Journal of Computers in Mathematics and Science Teaching, 1982, 1 (3), 9-10.

Moursund, D. Introduction to computers in education for elementary and middle school teachers. Eugene, Oregon: International Council for Computers in Education, 1981.

National Council of Teachers of Mathematics. An agenda for action. Reston, VA: Author, 1980.

Pratscher, S. Planning for microcomputers in the classroom. Journal of Computers in Mathematics and Science Teaching, 1981, 1 (1), 27-28.

Richman, E. Spotlight on computer literacy. New York: Random House, Inc., 1982.

Rogers, J. B. Computer education for colleges of education. New York: The Association for Computing Machinery, Inc., 1983.

Shane, H. G. The silican age II: Living and learning in an information epoch. Phi Delta Kappan, 1983, 65 (2), 126-129.

Sherwood, R. D., Conner, J. V. and Goldberg, K. P. Developing computer literacy and competency for preservice and inservice teachers. Journal of Computers in Mathematics and Science Teaching, 1981, 1 (2), 23-24.

Taylor, R. (Ed.) The computer in the school: Tutor, tool, tutee. New York: Teachers College Press, 1980.

Texas Education Agency. Computer literacy: K-Grade 12. Austin, TX: Author, 1983b.

Uhlig, G. Dimensions of technology literacy in techer education. Journal of Teacher Education, 1983, 34 (5), 2-5.

Walker, D. F. Reflections on the educational potential and limitations of microcomputers. Phi Delta Kappan, 1983, 65 (2), 103-107.

Zalewski, D. (Ed.) Microcomputers for teachers - With Application to mathematics and science, Topics for Teachers Series, Number 3. Bowling Green, Ohio: School Science and Mathematics Association, 1982.

2.6.4. A Case for Teachers to Acquire Computer Literacy

Trenton Cooper, Univ. of Arkansas-Pine Bluff, Pine Bluff, ARK

A position paper presented by AACTE Task force on Profiles of Excellence, lists five steps which they regard as necessary sequential items needed for the pursuit of excellence in teacher education. This paper discusses the first item listed, "the knowledge, skills, and attitudes to be guaranteed by graduation from Teacher Education." The assumption is that all AACTE member

institutions will put this in the form of a goal such as: "To convey the knowledge, skills, and attitudes that should typify the graduates from an approved teacher education program." It is further assumed that each member institution will implement actions to achieve this goal.

This paper takes the position that Computer Literacy is necessary for teacher educators who wish to intitiate actions that can adequately achieve the above goal. The task force members suggest four program components as a means of achieving this goal, they are: 1) general education, 2) academic specialization, 3) professional study in the disciplines undergirding pedagogy and 4) professional study. This paper uses these four program components to support the position of all teacher educators acquiring computer literacy.

Specifically, three sets of computing competencies are needed by teachers: 1) Those which all teachers must have regardless of their level or discipline, 2) Those needed only by the teacher of computing as subject, 3) Those needed for teachers who use computing to support or enhance instruction in subjects other than computing.

If teacher educators are to meet the challenge of preparing future teacher that are competent, then teacher educators will need all three sets of computing competencies. It is, therefore, essential for teacher educators to acquire computer literacy.

REFERENCES

Cicciarella, Charles F. "The computer in Physical Education: Its promise and threat." Journal of Physical Education, Recreation and Dance, (November/December, 1983), 1982.

Dick, Walter. "The preparation of Computer Educators, Educational Technology," XXIII, No. 9 (September, 1983), 38-39.

Glenn, Allen D., and Klassen, Danil L. "Computer Technology and the Social Studies," The Educational Forum XLVII, No. 2 (Winter, 1983, 209-16.

Harper, Dennis O., and Stewart, James H. Run: Computer Education. Monterey, CA: Brook/Cole Publishing Co., 1983.

Hasselbring, Ted. "Remediating Spelling Problems of Learning-Handicapped Students through the use of Microcomputers." Educational Technology, XXII, No. 4 (April, 1982), 31-32.

Kuykendall, Carol, "A Modest Agenda for Change," English Education, XLV, No. 3 (October, 1982), 149-156.

Lappan, Glenda and Winter, M. J. "A Unit on Slope Functions - Using a Computer in Mathematics Class," The Mathematics Teacher, 75, No. 2 (February, 1982), 118-122.

Poleger, Floyd D. "The Effectiveness of Microcomputers in Education" Southwest Educational Development Laboratory, 1983.

Rampy, Leah, White, Deborah, J. D. and Reckman, Saul. "Computers in the School: 21 Critical Issues for Policy Decisions," Educational Technology, XXIII, No. 8 (August, 1983), 20-23.

Schwartz, Lawrence. "Teaching writing in the age of The Word Processor and Personal Computers," Educational Technology, XXIII, No. 6 (June, 1983) 33-35.

Schwartz, Mimi. "Computers and the Teaching of Writing." Educational Technology, XXII, No. 11 (November, 1982).

2.6.5. Jousting with Jargon: Suggestions to Enhance Communication Between Computerists and Traditional Classroom Teachers

Jean Fincher, Highland Park Elementary School, Bartlesville, OK

In this transitional, every person-meets-machine phase of the computer revolution, language created by technocratic decree is supplanting to an unprecedented degree language created by evolutionary processes. Ambiguities of even the most commonly used terminology of computerists unnecessarily confuse classroom teachers and complicate their efforts to become comfortable with the use of computers in education. Personal and anecdotal accounts illustrate the implications of misunderstandings caused by such ambiguities. Until comprehension of the jargon of computerists becomes commonplace, the dilemma of this linguistic brouhaha can be resolved by the conscious substitution of that remarkably friendly interface, the English language.

This paper presents the perspective of an educator intimately familiar with the needs and aspirations of professional educators, specifically teachers and administrators. Specific suggestions for overcoming the linguistic impediments to effective computer implementation are offered, along with examples. The importance of overcoming the problems associated with too much computer jargon cannot be over emphasized. The dangers associated with jargon are best illustrated by the administrator who felt that there was no need to purchase software since faculty members could write their own using BASIC!

REFERENCES

Chicago teachers get computer training. Teaching and Computers, September, 1983, 8.

Kennedy, Don. Beginner's luck for the computer virgin. PC Magazine, July, 1983, 97-101.

Kennedy, Don. The computer virgin gets the word. PC Magazine, August, 1983, 379-386.

Papert, Seymour. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, 1980.

Petre, Peter D. Mass-marketing the computer. Fortune, October 31, 1983, 61-67

Schorr, Burt. Many schools buying computers find problems with using them. The Wall Street Journal, April 7, 1983, 29, 30.

Shaffer, Richard A. Courses in computer literacy beginning to draw bad marks. The Wall Street Journal, September 16, 1983, 33.

Vallee, Jacques. The Network Revolution: Confessions of a Computer Scientist. Berkeley: And/Or Press, 1982.

2.6.6. Computer Literacy Curriculum Guideline, K-12

Martha Kay Hedley, University of Alabama in Birmingham, Birmingham, AL

This curriculum guideline served as an incentive for other school systems and classroom teachers to develop an articulated computer literacy curriculum for their students in grades kindergarten through 12 just as articulated curricula have been developed in all other general education areas.

The guideline was subdivided by grade, topical area, performance objective, source, learning activity, evaluation, time, and subject area of integration. For each grade level a time guideline was given for two to six performance objectives involving topical concerns of the computer. The topics included were: (a) uses; (b) capabilities; (c) social; (d) historical; (e) vocabulary; (f) futuristics; (g) anatomy; (h) programming; and (i) algorithms. Each performance objective has a corresponding computing learning activity (either on-line or off-line), reference source, and a suggested student evaluation method. Specific curriculum area(s) are listed when computing topics can be integrated within the regular subject areas.

Three examples have been listed below from grades three, seven and eleven.

Grade	Topical Area	Performance Objective	Source	Learning Activity	Evaluation	Time	Subject Area of Integration
3	capabilities	1)Describe what a computer can and cannot do.	Bitter,p.38	1)Design a fantasy machine that will perform some task for us.	discussion	20mins	science
7	anatomy	1)Explain five components of a computer and uses.	Hedley lecture	1)Make model of computer. Draw diagram.	model and test	5-wks'	computer elective science
7	vocabulary	2)Define and use the basic operational symbols, commands and statements.	Cummins, p.30	2)Illustrate symbols, commands and statements.	matching test		
11	futuristics	2)Define artificial intelligence, robotics, cybernetics and other terms related to futuristics.	Hamm, pp.133-144	2)Draw system of future and write paper on how it works, added features, what it is used for, how much it costs, etc.	paper and test		

A complete curriculum guideline and reference listing can be obtained from the author.

REFERENCES

Adams, R. Clark & Poirot, James L. (1983). 40 easy steps to programming in basic and logo. Austin, TX: Sterling Swift.

Berger, Melvin. (1981). Computers in your life. New York: Thomas Y. Crowell

Bitter, Gary. (1983, October). The road to computer literacy. Electronic Learning, pp. 35-60.

BEST COPY AVAILABLE

Blank, Deborah. Educational computing post haste: A case study. Electronic Learning Facilitators, pp. 138-140. Bethesda, MD.

Clark, James & Drum, William. (1983). Basic programming: A structured approach. Cincinnati: South-Western.

Coburn, Peter. (1982). Practical guide to computers in education. Reading, MA: Addison-Wesley.

Cummins, Jerry & Kuehmann, Gene. (1983). Programming in basic. Columbus, OH: Charles E. Merrill.

Golden, Neal. (1981). Computer programming in the basic language. (2nd ed.). Atlanta: Harcourt Brace Jovanovich.

Horn, Carin & Poirot, James. (1981). Computer literacy--problem-solving with computers. Austin, TX: Sterling Swift.

Johnson, Mildred F. (1980, December). Computer literacy: What is it? Business Education Forum, p. 18.

Poirot, James. (1980). Computers and education. Austin, TX: Sterling Swift.

Rice, Jean. (1981). Computers are fun. Minneapolis, MN: Denison and Co.

Thomas, James. (1981). Microcomputers in the schools. Phoenix, AZ: Oryx Press.

2.6.7. Setting the TEMPO for Instructional Computing Courses: An Organized Set of Teaching Methods for Computer-Related Skills

LaJeane Thomas, Louisiana Tech University and Paul Ohme, Northeast Louisiana University

This paper discussed psychological considerations and some specific teaching methods that have been used effectively when teaching programming-related skills to preservice and inservice teachers. During the acquisition of these skills, this group (predominantly non-technical individuals) has been faced with obstacles such as fear of the computer and presentation of material. Suggestions for eliminating the fear of computers included:

1. motivate the students
2. provide opportunities for feelings of success
3. use humor
4. related vocabulary and examples to student background of experience
5. provide hands-on experiences
6. review and repeat concepts daily
7. observe for signs of fatigue, frustration, or signs that the student is feeling threatened.

When presenting computer-related materials, special consideration should be given to:

1. depth, pace, and sequence of material presented
2. vary the activities
3. progress from easy concepts to harder concepts
4. progress from concrete to abstract
5. involve as many senses in the learning process as possible
6. watch for a "saturation point"

When preparing an instructional sequence to present programming-related concepts, special consideration should be given to the teaching methods used. The following sequence of steps have been used when introducing commands, statements, or formatting techniques. They are:

1. introduce the statement or command
2. show the students the appropriate format and syntax for the statement
3. explain what the statement tells the computer to do
4. discuss what the screen output will be when the statement is executed
5. have the students type in a program that includes the statement as a part of a simple program
6. run the program and discuss the output

Reinforcement of the command or statements using the TEMPO method allowed the students opportunities to:

TRACE the execution of programs;
ELIMINATE ERRORS in sample programs;
MODIFY other's programs;
PROGRAM using a set of prescribed instructions;
OUTLINE a task and convert the outline into program statements.

These activities have helped the student to develop an understanding not only of the isolated programming statements and their syntax, but also an understanding of program design and how to achieve the desired output from their programming efforts. The TEMPO instructional procedures have been effective strategies, supplying the "external instructional events" necessary for supporting the internal learning processes needed for acquiring computer-related skills. Creative application of these methods along with establishment of a positive learning atmosphere has resulted in programming classes that have been not only fun but substantive.

REFERENCES

Estes, W. K. (1978). The information processing approach to cognition: A Confluence of metaphors and methods. In W. K. Estes (Ed.), Handbook of Learning and Cognitive Processes, Vol. 5, Human Information Processing. Hillsdale, New Jersey: Erlbaum.

Gagne, R. M. (1977). The conditions of Learning, 3rd edition. New York: Holt, Rinehart & Winston.

Gagne, R. M., W. Wager, and A. Rojas. (1981). Planning and authoring computer-assisted instruction lessons. Educational Technology, 21: 17-26.

Klatzky, R. L. (1980). Human memory: structures and processes, 2nd ed., San Fancisco: W. H. Freeman.

2.6.8. Who, What, When, Where and How: Important Considerations for Teacher Training in Microcomputer Technology

Mary M. Worth, Western New Mexico University, Silver City, NM

In teacher education, the importance of policy formation has almost been completely ignored prior to the selection of microcomputer technology. Yet, policy statements have acted as platforms for ensuing actions. Within teacher education research on policy development for microcomputer adoption and use has indicated four major areas of consideration. These are: curricular concerns, courseware/hardware issues, teacher training requirements, and community involvement. A policy statement reflecting each area should be written. For the first area, consideration should be given to goals and objectives of a computing curriculum, minimum computing competencies of students, and long and short-term methods of evaluation of the computing curriculum. For courseware/hardware, consideration should be given to: 1) evaluation/selection of software to meet curriculum needs; 2) evaluation of compatible software; 3) funding and support procedures; and 4) duplication and dissemination procedures For the third area (teacher training), guidelines should be written designating teacher competencies and corresponding inservice training necessary for teachin various levels of computing. For the final area, consideration should be given ensuring information dissemination among all interested community groups.

Studies have indicated several computing competency areas for all teachers. These areas are:

1. The capabilities of microcomputer systems.
2. The operation of a microcomputer system, including the appropriate terminology.
3. A survey of educational applications.
4. Some methods for evaluating software.
5. The ingredients of a programming language (probably BASIC).
6. An appraisal of the societal effects of the microcomputer as one element in the "age of information."

A secondary level of computing instruction should then be specialized to meet several criteria:

1. It must be determined by individual teachers based on their curriculum needs.
2. It must take into account specialized student needs.
3. There must be provision for adequate classroom instruction time where the teachers and students can be observed and receive feedback.
4. It must be an ongoing training experience with many opportunities for monitoring and discussion between teacher and trainer.

In preparing an inservice training sequence, Orlich (1983) has found nine key elements in the research literature indicating the most tested principles for successful inservice programs.

1. Teachers will benefit from those inservice programs which they helped plan.
2. School-oriented, school based programs are the most successful.
3. The goals of any inservice program should be clear, specific, and well-disseminated.
4. Individualized, small group training experiences may produce more positive and lasting results than programs which feature common activities for all participants.
5. If inservice training is to improve system operations then teachers' personal goals must coincide with those of the district.
6. Inservice training is more likely to be effective if adequate time is provided within the teachers' current work schedule.
7. Incentives must be identified for participants.
8. Involvement of the building principal in the program is critical for success.
9. Evaluation of any aspect of the inservice is important for it provides feedback so necessary adjustments can be made.

REFERENCES

Abernathy, S. Analysis of Microcomputer use and policy at New Mexico State University. Unpublished Doctoral dissertation, New Mexico State University, 1982.

Adams, J. Personal communication, July 19, 1983.

Cyrs, T. Personal communication, July 11, 1983.

Dede, C. The likely evolution of computer use in schools. Educational Leadership, 1983, 41, 22-23.

Denk, J. Personal communication, July 14, 1983.

Dryden, J. Personal communication, July 20, 1983.

Filliman, P. K. Guidelines for introducing microcomputers in the schools. Arithmetic Teacher, 1983, 30(6), 16-17; 56.

Komoski, K. P. Use these five ideas when drafting computer policies. The American School Board Journal, 1983, 170 (3), 30.

Lopez, A. Computer literacy for teachers: high school and university cooperation. Educational Technology, 1983, 21 (6), 15-18.

Lovell, P. Staff development for computer literacy. Educational Technology, 1983, 23 (3), 18-19.

Orlich, D. C. Some considerations for effective inservice education. Clearing House, 1983, 56, 197-202.

Ramey, L., White, D., & Rockman, S. Computers in the schools: 21 critical issues for policy decisions. Educational Technology, 1983, 23(8), 20-23.

Rottier, J. Receiving and individualizing inservice education. Education, 1983, 103, 274-277.

Sadowski, B. A model for preparing teachers to teach with the microcomputer. Arithmetic Teacher, 1983, 30(6), 24-25; 62-63.